

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Analýza a implementace věrnostního marketingového nástroje

Analysis and Implementation of the Affiliate Marketing Tool

Zadání bakalářské práce

Student: **Tomáš Lamacz**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Analýza a implementace věrnostního marketingového nástroje**
Analysis and Implementation of the Affiliate Marketing Tool

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce je implementovat věrnostní marketingový nástroj pro podporu prodeje jako samostatnou webovou aplikaci. Práce bude zaměřena na inovativnost a zkoumání stávajících řešení, hledání nových modelů a přístupů v dané oblasti a jejich praktickou implementaci.

1. Student provede analýzu existujících věrnostních marketingových nástrojů na podporu prodeje (českých i zahraničních). Analyzuje existující modely z pohledu implementace, obchodní strategie, zpoplatněných služeb a doplňkových funkcí.
2. Student provede návrh vlastního konceptu věrnostního marketingového nástroje v podobě generického modelu pro libovolný obsah.
3. Student provede analýzu, návrh a implementaci věrnostního marketingového nástroje ve vhodně zvoleném implementačním prostředí s použitím SQL nebo no-SQL databázových technologií.
4. V rámci implementace se student zaměří na pokročilé funkce vizualizace prezentovaného obsahu, responsivní design a další.
5. V závěru student provede srovnání dosažených výsledků s referenčními projekty a navrhne možnosti dalšího rozšíření.

Seznam doporučené odborné literatury:

- [1] GORMLEY, Clinton a Zachary TONG. Elasticsearch: the definitive guide. ISBN 1449358543.
- [2] SHKLAR, Leon. a Rich. ROSEN. Web application architecture: principles, protocols and practices. 2nd ed. Hoboken, NJ: Wiley, c2009. ISBN 047051860x.
- [3] SHIVAKUMAR, Shailesh Kumar. Architecting high performing, scalable and available enterprise web applications. ISBN 9780128022580.
- [4] WEERAWARANA, Sanjiva. Web services platform architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and more. Upper Saddle River, NJ: Prentice Hall PTR, c2005. ISBN 0131488740.


Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Radoslav Fasuga, Ph.D.**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2019




doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry


prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 30. dubna 2019

Tomáš Lomac
.....

Rád bych poděkoval vedoucímu mé bakalářské práce, Ing. Radoslavu Fasugovi, Ph.D, za poskytnutí důležitých rad, nápadů a připomínek, které mi s tvorbou práce pomohly.

Abstrakt

Tato bakalářská práce zkoumá problematiku věrnostních marketingových nástrojů. Jejím nejdůležitějším cílem je implementovat vlastní věrnostní marketingový nástroj formou webové aplikace, který bude sloužit pro podporu prodeje, evidenci obchodních případů a řízení prodejního týmu. První část práce se zabývá zkoumáním existujících obchodních modelů a aplikací. V další části popisují analýzu a implementaci vlastního nástroje pro evidenci obchodních smluv, kalkulaci provizí a administraci potřebné dokumentace. Pro implementaci bude použit programovací jazyk PHP a framework Nette.

Klíčová slova: PHP, Nette, marketing, věrnostní nástroj, obchodní zástupce, systém odměňování

Abstract

This bachelor thesis examines loyalty marketing tools. Its most important goal is to implement its own loyalty marketing tool in the form of a web application that will serve to support sales, business case records and sales team management. The first part deals with the research of existing business models and applications. In the next part I describe the analysis and implementation of my own tool for recording business contracts, commission calculation and administration of necessary documentation. The PHP programming language and the Nette framework will be used for implementation.

Key Words: PHP, Nette, marketing, affiliate tool, salesman, reward system

Obsah

Seznam použitých zkratk a symbolů	8
Seznam obrázků	9
Seznam tabulek	10
Seznam výpisů zdrojového kódu	11
1 Úvod	12
2 Popis aktuálního stavu	13
2.1 Druhy konverzí	13
2.2 Provize z doporučení on-line přes provizní systém	13
2.3 Provizní sítě	17
2.4 Oboustranně výhodné nabídky – doporučení od spokojeného klienta	19
2.5 Cashback portály	19
3 Specifikace zadání	21
3.1 Popis řešeného obchodního modelu	23
3.2 Popis technického řešení	24
3.3 Framework Nette	24
3.4 Databáze	26
3.5 Vzhled	26
3.6 Implementace aplikace ze strany klienta	26
4 Analýza	31
4.1 Doménový model	31
4.2 Koncept vzhledu uživatelského rozhraní	41
4.3 Implementace aplikace	44
5 Testování aplikace	49
5.1 Způsob testování	49
5.2 Průběh testování	49
5.3 Doporučení dalšího rozšíření	49
5.4 Průvodce aplikací	50
6 Závěr	51
Literatura	52

Seznam použitých zkratek a symbolů

PHP	– PHP: Hypertext Preprocessor
HTML	– Hyper Text Markup Language
SQL	– Standardized Query Language
MVC	– Model-view-controller
MVP	– Model-view-presenter
CSS	– Cascading Style Sheets
JSON	– JavaScript Object Notation
AJAX	– Asynchronous JavaScript and XML
API	– Application Programming Interface
CSV	– Comma-separated values

Seznam obrázků

1	ProAffil – nástěnka administrátora	14
2	Vzorový případ užití nástroje ProAffil	16
3	Katalog inzerentů na webu vivnetworks.com	18
4	Ukázka nákupu s Tipli	19
5	Případ užití: prodej	22
6	Případ užití: registrace partnera a přiřazení firmy partnerovi	22
7	Příklad rozdělení provizí	23
8	MVC architektura	25
9	MVP architektura	25
10	Tabulka vytvořená pomocí Metronic Datatables	27
11	Ukázka použití komponenty SweetAlert2 pro validaci formuláře	29
12	Konceptuální diagram	32
13	Relační diagram	33
14	Koncept vzhledu nástěnky	41
15	Koncept vzhledu seznamu produktů	42
16	Koncept vzhledu detailu produktu	42
17	Koncept vzhledu editace produktu	43
18	Flash zpráva oznamující úspěšnou změnu hesla	45

Seznam tabulek

1	Příklad rozdělení provizí	23
2	Srovnání řešení PHP a ASP .NET [12]	24
3	Datový slovník tabulky Faktura	35
4	Datový slovník tabulky Firma	35
5	Datový slovník tabulky Forma_spoluprace	35
6	Datový slovník tabulky Moje_firma	35
7	Datový slovník tabulky Nabidka	36
8	Datový slovník tabulky Objednavka	36
9	Datový slovník tabulky Platba	36
10	Datový slovník tabulky Polozka_objednavky	37
11	Datový slovník tabulky Prodejce	37
12	Datový slovník tabulky Produkt	37
13	Datový slovník tabulky Produkt_vlastnost	37
14	Datový slovník tabulky Provize	38
15	Datový slovník tabulky Schuzka	38
16	Datový slovník tabulky Smlouva	38
17	Datový slovník tabulky Tarif	38
18	Datový slovník tabulky Typ_smlouvy	38
19	Datový slovník tabulky Typ_udalosti	39
20	Datový slovník tabulky Typ_vlastnosti	39
21	Datový slovník tabulky Ucet_objednavky	39
22	Datový slovník tabulky Ucet_provize	39
23	Datový slovník tabulky Udalost	39
24	Datový slovník tabulky Udalost_uzivatel_zobraz	40
25	Datový slovník tabulky Uzivatel	40
26	Datový slovník tabulky Vlastnost_produkту	40

Seznam výpisů zdrojového kódu

1	Ukázka použití komponenty SweetAlert2	28
2	Výpis flash zpráv do šablony	45

1 Úvod

Cílem mé bakalářské práce je navrhnout a naimplementovat věrnostní marketingový nástroj pro podporu prodeje jako samostatnou webovou aplikaci.

V první, teoretické, části se zaměřuji na stávající, české i zahraniční obchodní modely na podporu prodeje. Jsou zde vyjmenovány druhy konverzí, za které lze získat odměnu. Je zde také detailně popsán princip použití provizního nástroje ProAffil. Dále v této části popisuji princip fungování provizních sítí, jako je web vivnetworks.com nebo český web dognet.cz. Zmiňuji se i o oboustranně výhodných nabídkách za doporučení služby nebo produktu a popisuji, jak fungují tzv. cashback portály, které vrací peníze zákazníkům z nákupů v internetových obchodech.

Další, prakticky zaměřená část, se skládá z popisu obchodního modelu navrhovaného informačního provizního systému pro obchodní zástupce firmy, detailního popisu technického řešení, technologií a postupů při implementaci systému. Je zde popsán například přístup k databázi, odesílání emailů, generování PDF souborů, vytváření formulářů, nebo zpracování AJAX požadavků.

Předposlední část je zaměřena na testování naimplementovaného systému. Je zde popsáno, proč je nezbytné otestovat aplikaci před jejím nasazením, dále průběh a výsledky testování, z nichž vyplynuly možnosti pro budoucí rozšíření aplikace.

Závěr bakalářské práce je shrnutím dosažených výsledků.

2 Popis aktuálního stavu

Tato kapitola je věnována zavedeným obchodním modelům, které umožňují lidem vydělávat peníze formou provizí. Zaměřuje se na jejich princip a způsob využití. Popisují je na existujících nástrojích. K popisu obchodních modelů jsem vybral známé a dostupné marketingové nástroje z České republiky i ze zahraničí.

2.1 Druhy konverzí

Provize se běžně získávají za tyto konverze:

1. Nákup na internetu
2. Přivedení zákazníka k nějaké službě
3. Sjednanou smlouvu - např. pojištění, půjčka
4. Přivedení dalšího affiliate partnera
5. Z prokliků

Provize může být vyplacena v peněžní částce nebo v kreditech, které může partner použít třeba k nákupu u poskytovatele provize.

Pro měření konverzí se obvykle používají cookies, které se ukládají v prohlížeči. Cookies jsou uloženy do počítače při kliknutí na odkaz po dobu, která je nastavena v provizním programu. Po tuto dobu jsou měřeny konverze, tedy návštěvy konverzních cílů.

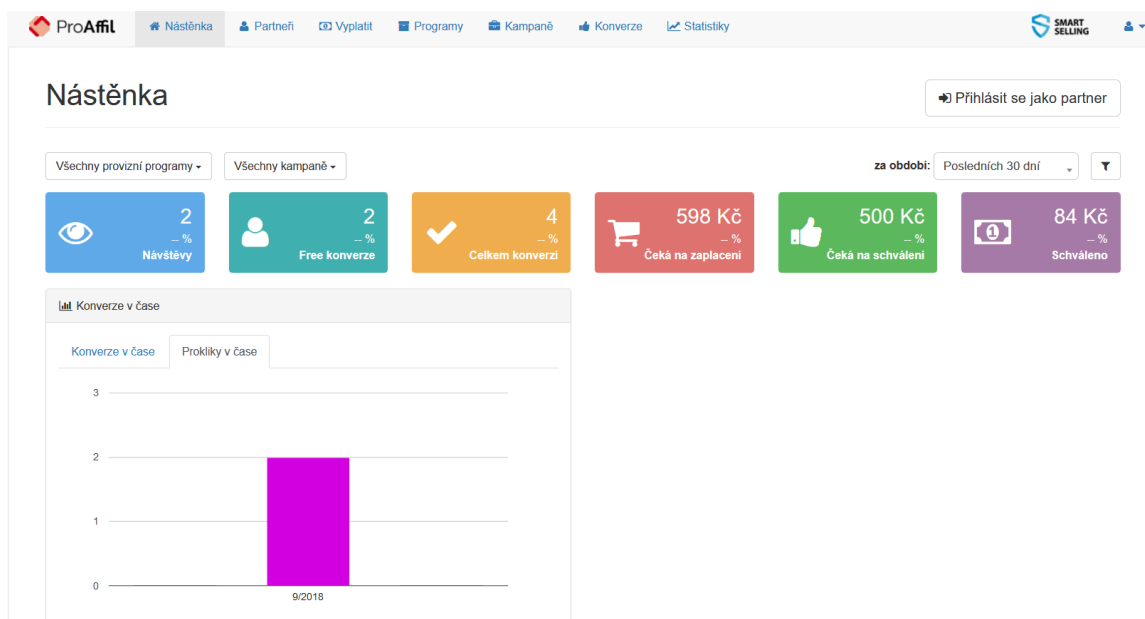
2.2 Provize z doporučení on-line přes provizní systém

Obchodník může provozovat své vlastní programové řešení pro provizní systém, příkladem je řešení od Manutan.cz [1]. Dále se může spolehnout na existující služby určené pro správu provizních kampaní. Mnoho z nich je velmi propracovaných, a tak je tato možnost pro obchodníka často výhodnější.

2.2.1 ProAffil

Příkladem takové služby je jednoduchá aplikace ProAffil od české firmy SmartSelling a.s., kterou je možné si vyzkoušet zdarma v demo verzi na webu [2]. Prostředí demo aplikace hodnotím jako přehledné. Nevýhodou tohoto demo je, že nelze data ukládat, a tak je možné jen zobrazovat ukázková data. Nevýhodou samotné aplikace je, že pro její chod je nutné mít od firmy SmartSelling a.s. aktivní také její nástroje SmartEmailing a Fapi.

Po přihlášení jako administrátor se zobrazí nástěnka s přehledem, ukazující počet návštěv, konverzí a odměn. Je možné nastavit, ze kterých provizních programů a kampaní, či za jaké



Obrázek 1: ProAffil – nástěnka administrátora

časové období se mají tato čísla zobrazovat. Pod tímto přehledem se nachází jednoduchý graf zobrazující konverze a prokliky v čase.

Na záložce „Partneři“ vidíme přehled, který nabízí také přidávání a úpravy partnerů. O partnerech jsou ukládána tato data: zda je VIP, email, affiliate ID (podle tohoto se tvoří odkaz pro partnera), jméno, telefon (nepovinné), provizní program a datum registrace.

Záložka „Vyplatit“ umožňuje jednorázově vyplatit partnera částkou, kterou zadáme do pole „Částka“. Nástroj neumožňuje nastavit vyplácení automaticky, např. měsíčně.

V přehledu programů, do kterých se dále zařazují jednotlivé kampaně, máme o každém programu informaci zda je aktivní, jaký je jeho název, identifikátor a kdo je jeho manažerem. Je zde samozřejmě možnost přidávat a upravovat programy.

U kampaní je evidován provizní program, název, popis, odkaz na stránku s bannery, zda je provize počítána v procentech nebo je to fixní částka, hodnota provize, platnost doporučení (počet dní od kliknutí na odkaz, do kterého je třeba provést konverzi) a délka garance vrácení peněz ve dnech (konverze může být zamítnuta například z důvodu storna objednávky, po uplynutí této doby je konverze automaticky schválena).

V přehledu konverzí je možné konverze schvalovat, zamítat, duplikovat, editovat, ale také vytvářet nové konverze. Jsou zobrazeny informace, jako je datum vytvoření, schválení, status, název kampaně, variabilní symbol, výše odměny a partner. Pro větší přehlednost je můžeme také filtrovat podle jejich statusu a podle kampaně. Také v nich můžeme vyhledávat.

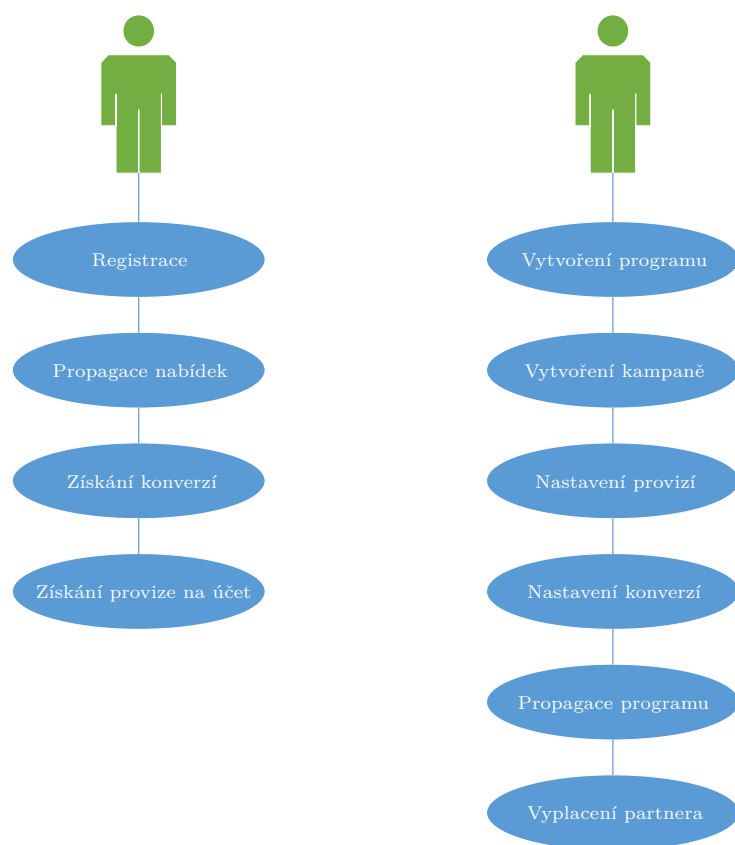
Na poslední záložce se statistikami je možné sledovat, jak si vedou jednotliví partneři a kampaně v počtu návštěv, konverzí, obratu a provizi. Podle těchto čísel je možné také data řadit.

Cena nástroje ProAffil se odvíjí od zvoleného tarifu. V nabídce je měsíční tarif za 450 Kč měsíčně a výhodnější roční tarif za 4500 Kč ročně. Dále je možné se domluvit na jiné ceně, pokud má zákazník individuální přání [3].

2.2.2 Příklad využití nástroje ProAffil

Pro přiblížení uvedu konkrétní příklad využití.

Osoba, která prodává online kurzy o podnikání na svém webu (pomocí prodejního formuláře Fapi), vytvoří provizní program „Podnikání na internetu“. V tomto programu vytvoří kampaň pro každý ze svých online kurzů, například „Jak vydělat na Amazonu“ a „Jak vybudovat úspěšný eshop“. Cena každého kurzu je 1500 Kč. U první kampaně nastaví hodnotu provize na 10 %, u druhé nastaví provizi jako fixní částku 150 Kč (tím dojde ke stejnému výsledku, jako u první kampaně). Má také možnost nastavit vyšší hodnotu provize pro VIP partnery, např. 200 Kč. Dále má možnost nastavit hodnoty konverzí za produkty (kurzy) zdarma. Na svém webu pak propaguje tyto kampaně například pomocí bannerů. Když se návštěvník dostane přes odkaz z webu na stránky kampaně na ProAffilu, uvidí popis kampaně, ve kterém by mělo být uvedeno co se prodává a jaká je provize. Podle toho se pak návštěvník rozhodne, zda se chce stát partnerem pro tuto kampaň. U kampaní je také třeba nastavit z jakého prodejního formuláře se mají započítávat konverze a za jakou konkrétní položku. Pokud se návštěvník rozhodne pro zapojení do programu, musí si založit partnerský účet, pokud jej ještě nemá. Ve svém účtu pak vidí odkazy, které použije pro doporučení třeba na svém webu, či na sociálních sítích. Osoba, která se přes daný odkaz dostane na stránku produktu, musí provést konverzi do doby, která je nastavena u kampaně v poli „platnost doporučení“, aby se konverze započítala a byla partnerovi připočtena provize.



Obrázek 2: Vzorový případ užití nástroje ProAffil

2.2.3 Další poskytovatelé provizních řešení









ProAffil není jedinou českou aplikací svého typu, jeho českým konkurentem je nástroj AffilBox. Ten také nabízí demo k vyzkoušení zdarma. Je nabízen ve dvou tarifech, kde cena levnějšího činí 8458 Kč s DPH za rok [4]. V levnějším tarifu je možno vytvořit maximálně 5 kampaní. Z demo verze je patrné, že vzhled webu není responzivní, i tak je ale podle mého názoru dostatečně přehledný. Jeho velkou konkurenční výhodou je možnost snadného propojení například s e-shop systémem od Shoptet. Cena propojení je v tomto případě jednorázově 1990 Kč [5]. Shoptet však nabízí vedle tohoto řešení i svůj vlastní modul, který je výrazně levnější, jednorázová cena je jen 3990 Kč [6]. Nevýhoda je pak hlavně v tom, že je systém vázán na konkrétní e-shop. Dražší tarif od AffilBoxu za roční poplatek 24188 Kč nabízí navíc neomezené množství kampaní, odkazování bez parametru (uživatel nepozná, zda jde o affiliate odkaz) a automatické nastavení jazyka dle lokality partnera [4].

Mezi populární zahraniční systémy patří například Post Affiliate Pro, který rovněž podporuje odkazování bez parametru, a to již v nejlevnějším tarifu. Cena v tomto případě začíná na 97\$ měsíčně [7].

2.3 Provizní sítě

Pro usnadnění vyhledávání provizních programů nám slouží tzv. provizní sítě. Provizními sítěmi jsou webové stránky, které fungují jako katalogy inzerentů, pro něž se publisher může stát partnerem. Publisherem je ten, kdo propaguje služby například na webech s velkou návštěvností, na Instagramu, na YouTube, nebo na tzv. cashback portálu. Jako příklad provizní sítě uvádím web vivnetworks.com, na kterém je inzerováno 275 kampaní. Všechny provizní kampaně fungují přes provizní systém CJ.com, na kterém si partner musí vytvořit účet.

Druhým příkladem je web dognet.cz, který nabízí 309 kampaní. Na rozdíl od předchozího funguje web zároveň jako komplexní provizní systém, takže inzerenti spravují své kampaně přímo na tomto webu.

<div>  <div> Úvod Pro partnery Akce inzerentů Katalog Pro inzerenty Konference Blog Napsali o nás Kontakty <div> CZ </div> </div> </div>					
Název programu	Velikost provize	Délka cookie	Detaily	Máte účet v CJ?	
 123zlavy.sk Slevové portály, Slovenský	4%	45 dní	Detail programu	Zapojit se	
 4camping.cz Cestování, Sport	5-8 %	30 dní	Detail programu	Zapojit se	
 51015kids.cz Český, Oblečení a móda	8%	30 dní	Detail programu	Zapojit se	
 Actionzazitky.sk Slovenský, Sport	4 - 6%	45 dní	Detail programu	Zapojit se	
 Adrop.cz Český, Sport, Zážitky a ubytování	12%	45 dní	Detail programu	Zapojit se	
 Adrop.sk Slovenský, Sport, Zážitky a ubytování	10 - 12%	45 dní	Detail programu	Zapojit se	
 Ageo.cz Český, Dětsá hračky, Kreslení a dělostroj	4,5 - 6%	45 dní	Detail programu	Zapojit se	

Obrázek 3: Katalog inzerentů na webu vivnetworks.com



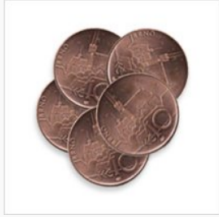
2.4 Oboustranně výhodné nabídky – doporučení od spokojeného klienta

Jako příklad uvádím nabídku aplikace SmartEmailing, nástroje pro rozesílku hromadných emailových kampaní, která nabízela v roce 2016 pro své klienty odečtení 500 Kč z faktury za přivedení nového klienta přes vlastní odkaz vygenerovaný v aplikaci, kde nový klient získal také 500 Kč [8]. Pro budoucího klienta tak bylo výhodné použít při registraci tento odkaz. Podmínkou bylo také, že klient zůstane zákazníkem minimálně po dobu dvou měsíců. Díky tomu, že aplikace slouží k rozesílce hromadných emailů, mohli klienti vkládat své odkazy na doporučení přímo do emailů, například do patičky. Z takové nabídky pak profituje firma, stávající spokojený zákazník a nový zákazník. Nabídky fungující na tomto principu používají často také různé banky, příkladem je spořicí účet ING Konto od ING, kde nový i stávající klient dostane 300 Kč, pokud nový klient vloží na konto během prvních třiceti dní alespoň 5000 Kč [10].

2.5 Cashback portály

Cashback portály jsou webové stránky, které vrátí zákazníkovi část peněz, které v internetovém obchodě utratil. Zákazník obvykle musí nákup provést přes odkaz z portálu.

Mezi nejznámější portály u nás patří web Tipli.cz, který se sice pyšní tím, že spolupracuje s více než 1800 obchody, avšak odměny lze (ke dni 21. 9. 2018) získat jen v 793 obchodech [9]. Mezi nimi jsou nejen české, ale i známé zahraniční obchody, jako AliExpress nebo Booking.com. Mnoho zapojených obchodů na tomto portálu však nenabízí peníze zpět, ale jen zde vystavuje slevy a kupóny. Lákadlem je také bonus 100 Kč pro nové uživatele, kde je podmínkou získání 100 Kč zpět během prvních 100 dní od registrace. Portál nabízí také doplněk do prohlížeče, který upozorní při návštěvě zapojeného obchodu, že zde platí cashback. Další výhodou je vyplácení odměn už od 1 Kč.

			
Na letní dovolenou potřebujete výhodně koupit nové plavky na MALL.cz za 1899 Kč.	Na Tipli.cz vyhledáte MALL.cz a prokliknete se do obchodu.	Přidáte plavky do košíku a nakoupíte jako obvykle. Zbytek je na nás.	Automaticky zaregistrujeme na účet vaší odměnu ve výši 76 Kč.

Obrázek 4: Ukázka nákupu s Tipli

Dalším oblíbeným portálem je PlnáPeněženka.cz, ten vrací většinou stejně velké nebo menší procento. Spolupracuje také s AliExpress, ale odměna je maximálně 5 %, zatímco na Tipli.cz je to až 10 %. Pro nové uživatele je zde bonus maximálně 60 Kč. Aby jej mohl uživatel získat celý, musí provést během prvních třiceti dní nákupy ve třech různých obchodech. Také nabízí doplněk

do prohlížeče, který upozorní na zapojený obchod při jeho navštívení, navíc je za jeho instalaci připsáno uživateli na účet 10 Kč. Web umožňuje získat peněžní odměnu až 6000 Kč také za doporučení kamarádovi, což Tipli.cz umožňuje také, avšak odměna u Tipli.cz činí maximálně 4500 Kč.

Z cashback portálů profitují tři strany. Tou první je samotný portál, který získává provizi ze zprostředkování obchodu. Dále jsou to zapojené firmy, protože portál jim poskytuje reklamu a zprostředkovává prodeje. Většina zapojených firem na Tipli.cz spolupracuje s provizními sítěmi, samotnému portálu Tipli.cz platit nemusí. Tipli funguje v provizních sítích jako publisher a přes tyto sítě získává provize z prodejů. Třetí stranou je zákazník, který díky této reklamě nakoupí v obchodě dané firmy, protože získat peníze z nákupu zpět je pro něj výhodné.

Shrnutí

Prošli jsme si, za jaké činnosti jsou běžně lidé na internetu odměňováni a jak velkou odměnu mohou získat. Také víme, co z toho získá odměňovatel. Pokud je systém odměn výhodný pro všechny strany účastníků, je to jednoznačně velmi užitečný marketingový nástroj pro získání nových klientů.

3 Specifikace zadání

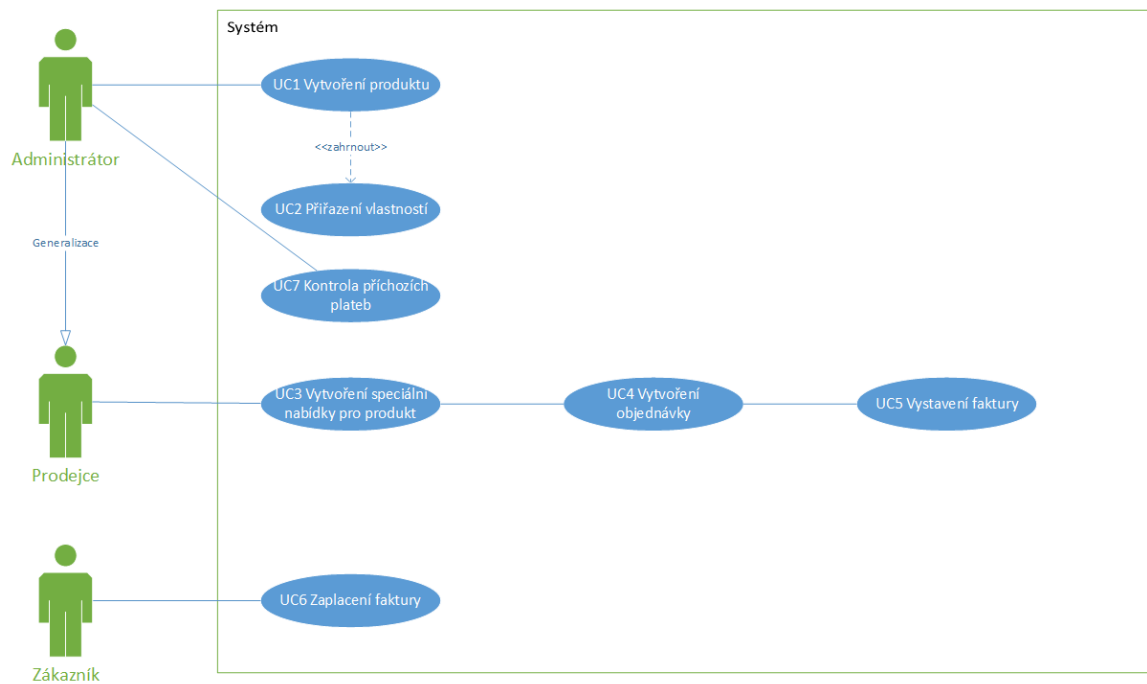
Navrhuji informační provizní systém pro obchodní zástupce firmy.

Se systémem budou pracovat obchodní zástupci (partneři), kteří uzavírají obchody s různými firmami (vytvářejí objednávky). Z těchto objednávek získají provize. Zástupci mohou nabídnout také vyšší, či nižší cenu produktů, a to tak, že k produktu vytvoří speciální nabídku. V takovém případě se stejná procenta, o které bude cena zvýšena/snížena, přičtou/odečtou od částky provize ze standardní ceny. Systém bude dále umožňovat zástupcům plánovat schůzky s firmami v plánovacím kalendáři. Zástupci uvidí v systému seznam firem, se kterými mohou spolupracovat.

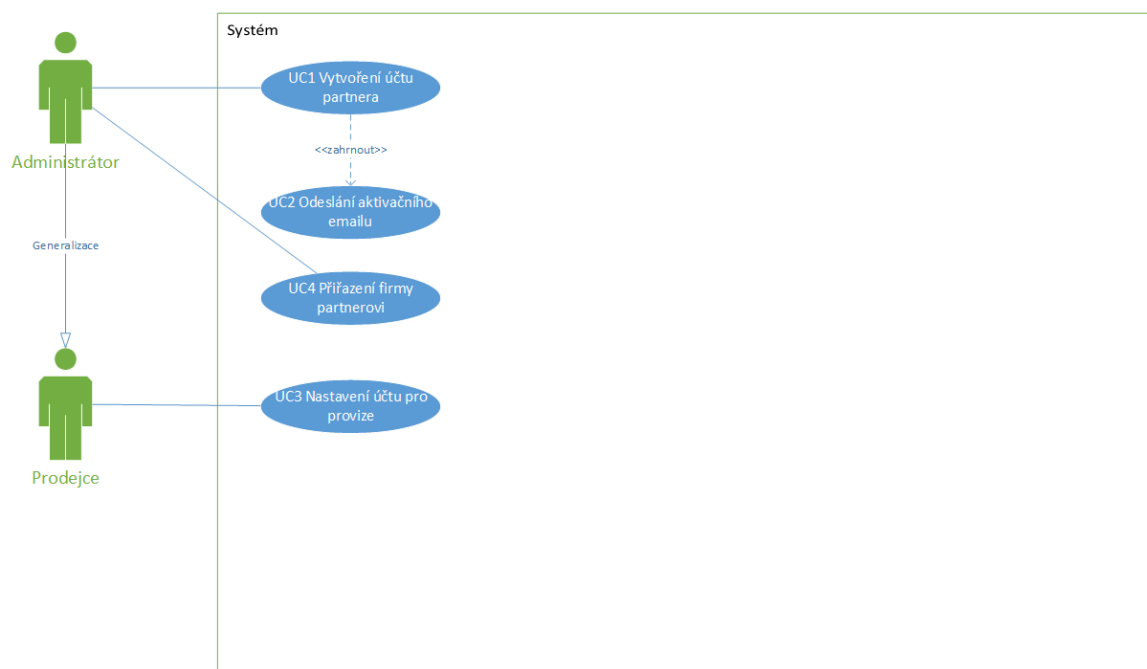
Zástupci budou mít přiřazen svůj tarif podle toho, kolik kusů produktů prodají za určitou dobu. Mohou vytvářet účty také svým podřazeným prodejcům, kteří mu pomohou s prodejem produktů. Může každému vybrat, kterým firmám je smí prodávat a v jakém časovém rozmezí. K vytvoření účtu potřebuje vyplnit formulář se základními údaji, včetně emailové adresy a výši provize. Poté systém automaticky odešle novému uživateli emailovou zprávu s odkazem pro dokončení registrace. Po kliknutí na tento odkaz novým prodejcem bude potřeba vytvořit heslo pro přihlášení.

Dále bude se systémem pracovat administrátor. Administrátor je nadřazený všem zástupcům a může navíc vytvářet a editovat jednotlivé produkty, jejich běžnou cenu, minimální a maximální cenu v případě speciální nabídky a další vlastnosti. Administrátor bude mít přístup k seznamu přijatých plateb nastaveného bankovního účtu.

Obrázky 5 a 6 znázorňují možné případy užití aplikace.



Obrázek 5: Případ užití: prodej



Obrázek 6: Případ užití: registrace partnera a přiřazení firmy partnerovi

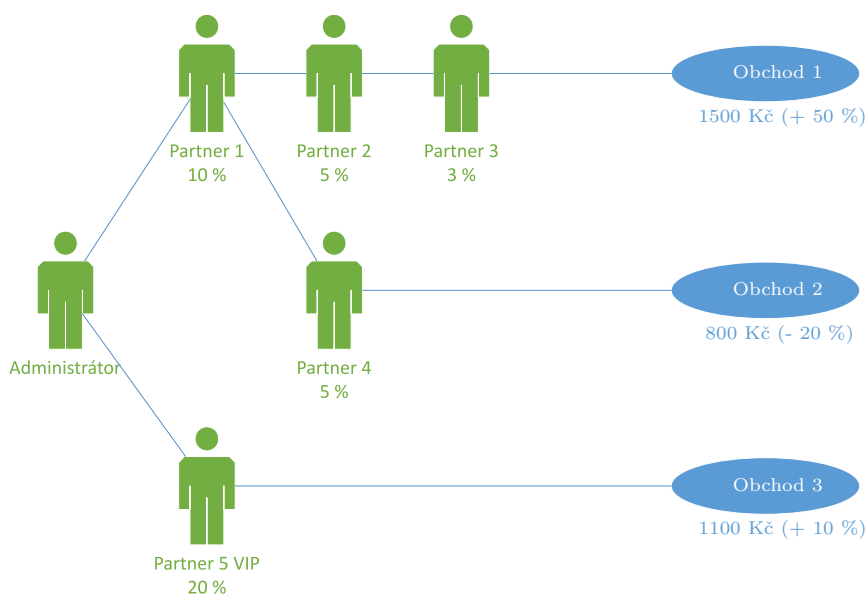
3.1 Popis řešeného obchodního modelu

Obchodní model pro tuto aplikaci popisují na modelovém příkladu obchodní situace.

Partneři budou vytvářet obchody na produkt „Reklama“ se základní cenou stanovenou administrátorem 1000 Kč. Minimální cena je 800 Kč a maximální cena je 1500 Kč. Výchozí nastavení provizí a hierarchie uživatelů znázorňuje obrázek 7. V příkladu je každý z obchodů proveden jednou. Všichni aktéři používají plný tarif se sazbou 100 % (konečná částka provize je tedy stejná jako částka provize ve sloupci „Provize z konečné ceny“).

Tabulka 1: Příklad rozdělení provizí

		Provize ze základní ceny	Provize z konečné ceny
Partner 1	Obchod 1	$10 - (5 + 3) = 2 \%$	$10 \% * 1,5 - 7,5 \% = 7,5 \% (75 \text{ Kč})$
	Obchod 2	$10 - 5 = 5 \%$	$10 \% * 0,8 - 4 \% = 4 \% (40 \text{ Kč})$
Partner 2	Obchod 1	$5 - 3 = 2 \%$	$5 \% * 1,5 - 4,5 \% = 3 \% (30 \text{ Kč})$
Partner 3	Obchod 1	3 %	$3 \% * 1,5 = 4,5 \% (45 \text{ Kč})$
Partner 4	Obchod 2	5 %	$5 \% * 0,8 = 4 \% (40 \text{ Kč})$
Partner 5 VIP	Obchod 3	20 %	$20 \% * 1,1 = 22 \% (220 \text{ Kč})$



Obrázek 7: Příklad rozdělení provizí

3.2 Popis technického řešení

Systém bude fungovat ve formě webové stránky, aby uživatelé nemuseli nic instalovat do svých zařízení. Vzhled bude přizpůsoben pro zařízení s různě velkými obrazovkami.

Systém bude naprogramován v jazyce PHP, protože tento jazyk je multiplatformní – na rozdíl od ASP .NET tak odpadá nutnost použít jej na platformě Microsoft IIS, kde je třeba platit licenční poplatek. Můžeme jej zprovoznit například na serveru se systémem Linux za nižší náklady. Podrobnější srovnání je uvedeno v tabulce Tabulka 2.

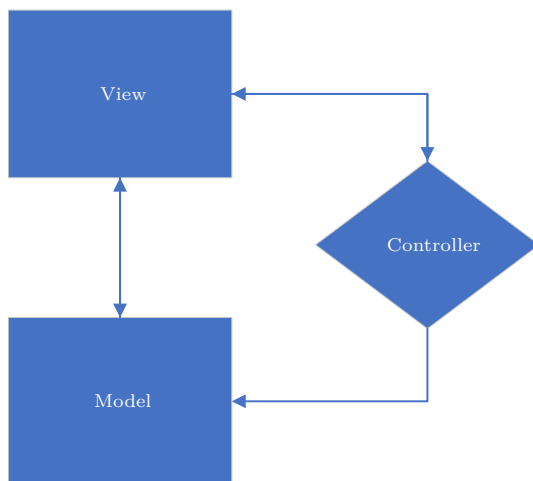
Tabulka 2: Srovnání řešení PHP a ASP .NET [12]

	PHP	ASP .NET
Typ	Skriptovací jazyk na straně serveru	Framework s podporou jazyků .NET
Podpora	Malé až střední projekty	Střední a velké projekty
Cena software	Zdarma	Zdarma
Cena platformy	Zdarma	S licenčním poplatkem
Komunita	Velká	Malá

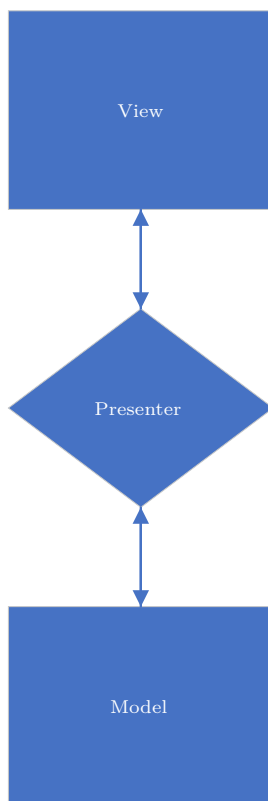
3.3 Framework Nette

Protože systém bude složitější, bude vhodné využít pro usnadnění práce a lepší přehlednost již hotový Framework (soubor knihoven). Pro jazyk PHP jich existuje celá řada. Pro vývoj tohoto systému jsem zvolil volně dostupný Framework Nette, který je populární hlavně v České republice. Autorem je český vývojář David Grudl, a také o tomto frameworku existuje české fórum. U Nette je dbáno na bezpečnost, používají se například technologie pro eliminaci výskytu Cross-Site Scripting útoků. Pro tento Framework existuje také spousta doplňků. Nette verze 2.4 funguje na PHP od verze 5.6 a je kompatibilní s verzí 7.3. [15]

Framework Nette je postaven na architektuře Model-View-Presenter, která je variantou známé architektury Model-View-Controller. Rozdíl je, že vrstva controller je nahrazena presenterem. Ten se stará na rozdíl od controlleru také o naplnění dat do vrstvy view a volá presenter, když obdrží události z view. [16]



Obrázek 8: MVC architektura



Obrázek 9: MVP architektura

Nette dále obsahuje například knihovnu pro tvorbu formulářů Nette Forms, dále systém pro tvorbu šablon Latte a vrstvu pro práci s databází.

3.4 Databáze

Jako databázový server jsem zvolil MySQL, protože je to jeden ze systémů, které Nette plně podporuje. Dalším důvodem je, že jde o open source a je dostupná na většině webhostingů. Pro správu databáze budu používat grafické rozhraní phpMyAdmin, které se v kombinaci s MySQL používá běžně.

3.5 Vzhled

Protože systém má fungovat na různých zařízeních od stolních počítačů až po mobilní telefony, bude třeba vytvořit responzivní design webu, aby se přizpůsobil různým velikostem obrazovek. Abych nemusel psát celý CSS stylpis od základu, rozhodl jsem se použít pro stylování uživatelského rozhraní již hotovou šablonu. Zvolil jsem moderní šablonu Metronic od Keenthemes [13]. Tato šablona je velmi populární a je určena zejména pro administrační webové rozhraní. V době psaní této práce obsahuje 12 různých verzí, ze kterých je možno vycházet. Jednotlivé části těchto verzí se dají kombinovat. Ve své práci vycházím hlavně z verze „Demo12“. Šablona je postavena na open-source knihovně Twitter Bootstrap (verze 4.1.3), která se skládá z mnoha komponent zejména pro stylování webů do responzivní podoby a pluginů založených na frameworku jQuery.

Samotná šablona je připravena na použití velkého množství rozšíření a ikon od nejrůznějších autorů, ale obsahuje také vlastní doplňky, jako je například Metronic Datatables.

3.5.1 Bootstrap 4.1.3

Bootstrap je open-source framework pro vývoj webových aplikací pomocí HTML a CSS. Je modulární a sestává se z LESS stylpisů pro jednotlivé komponenty. Díky použití jazyka LESS lze snadno provádět změny nastavení šablony. Framework už od jedné z prvních verzí podporuje tzv. grid-systém, který funguje pomocí zlomových bodů (break-points) pro různé šířky prohlížeče. Pomocí přiřazování tříd jednotlivým elementům se pak určuje, jak širokou část prohlížeče mají zabírat při jednotlivých šířkách okna prohlížeče [14].

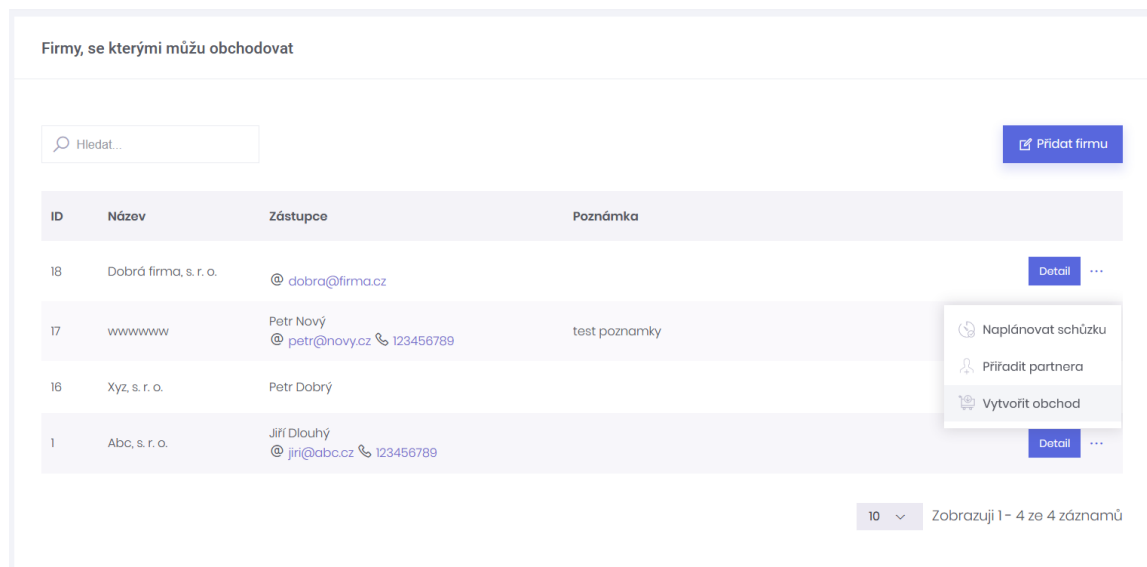
3.6 Implementace aplikace ze strany klienta

Na straně klienta aplikace je potřeba řešit také jednoduchou logiku, například naplnění dat do kalendáře pro zobrazování schůzek nebo tabulek. Tato logika bude naprogramována pomocí jazyka JavaScript s použitím knihovny JQuery a jejími rozšířeními.

3.6.1 Metronic Datatables

Metronic Datatables je komponenta, která umožňuje pokročilou tvorbu tabulek. Podporuje například řazení dat na straně klienta (včetně vlastní callback funkce, která definuje algoritmus pro řazení) nebo serveru, vyhledávání, filtrování a stránkování na straně klienta nebo serveru. Potřebujeme-li hodnotu buňky před výpisem upravit, nebo chceme mít v buňce místo prostého

textu HTML kód, máme k dispozici metodu `render`, která vrací výslednou podobu buňky. Jako zdroj dat pro naplnění tabulky lze použít datové pole, JSON soubor získaný ze serveru, nebo lze tabulku inicializovat přímo z kódu HTML. Pro svůj systém načítám data z JSON souboru. [17]



ID	Název	Zástupce	Poznámka
18	Dobrá firma, s. r. o.	@ dobra@firma.cz	
17	wwwwww	Petr Nový @ petr@novy.cz 123456789	test poznámky
16	Xyz, s. r. o.	Petr Dobrý	
1	Abc, s. r. o.	Jiří Dlouhý @ jiri@abc.cz 123456789	

Obrázek 10: Tabulka vytvořená pomocí Metronic Datatables

3.6.2 FullCalendar 3.9.0

Tuto komponentu kalendáře používám jen pro zobrazení schůzek, pro jejich vytváření a úpravy je nutné využít samostatný formulář, nicméně komponenta nabízí také možnost tzv. drag-n-drop, která umožňuje přesunovat události mezi jednotlivými hodinami nebo dny. Pro naplnění dat lze použít pole, JSON soubor, nebo vlastní funkci pro generování událostí programově. Zdrojů dat může být i více než jeden. Pro svůj systém načítám data opět z JSON souboru. [18]

3.6.3 Bootstrap TouchSpin 4.2.5

Tato formulářová komponenta přidá k vstupnímu poli, do něž se zadávají čísla, tlačítka pro přidávání a ubírání hodnoty pole. Lze nastavit například minimální a maximální hodnotu, dále hodnotu, o kterou se bude hodnota pole měnit, počet desetinných míst, postfix a prefix. Pokud uživatel zadá neplatnou hodnotu, hodnota pole se po kliknutí mimo pole automaticky přepíše na nejbližší platnou hodnotu. Pokud ale uživatel po zadání neplatné hodnoty stiskne klávesu Enter, odešlou se tato neplatná data. Je proto stále třeba kontrolovat tato data zvlášť. [19]

3.6.4 Datepicker 4

Komponenta slouží ke zjednodušení zadávání datumů a časů do textového pole. Uživatel nemusí psát datum pomocí klávesnice, stačí jen vybrat datum nebo čas kliknutím do kalendáře, který se zobrazí po kliknutí do textového pole v jeho blízkosti. [20]

3.6.5 Bootstrap Select

Bootstrap Select slouží pro přizpůsobení rozbalovacích nabídek ve formulářích, nahrazuje nativní rozbalovací nabídky webového prohlížeče a je možno ji jednoduše nastýlovat. Oproti nativní rozbalovací nabídce nabízí více funkcionality. Mezi zajímavé funkce této komponenty patří možnost výběru více hodnot (přidáním atributu **multiple**) včetně možnosti omezení počtu vybraných hodnot, vyhledávání v možnostech (při nastavení atributu **data-live-search** na hodnotu „true“), možnost nastýlovat jednotlivé možnosti zvlášť, vložení vlastního HTML obsahu do možnosti (pomocí atributu **data-content**), nebo výběr všech možností pomocí tlačítka. [21]

3.6.6 SweetAlert2

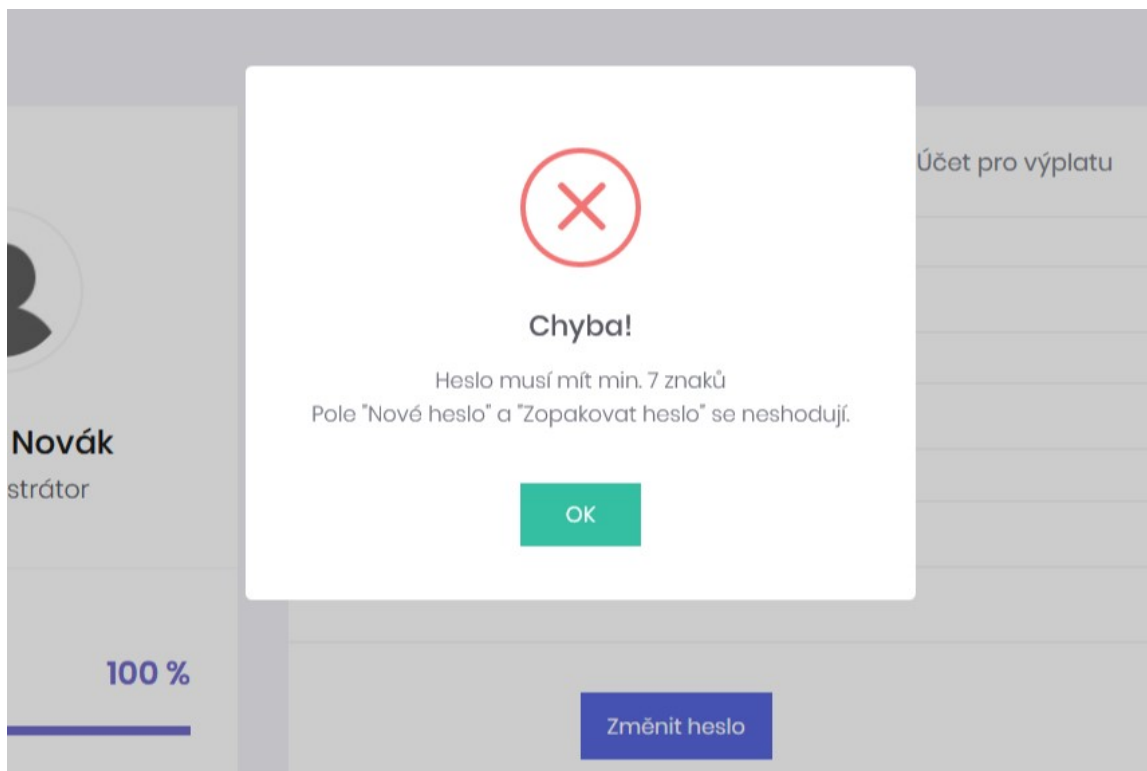
SweetAlert2 je komponenta, která slouží jako náhrada za klasický vyskakovací box v prohlížeči. Používá vlastní styl a zobrazuje se na rozdíl od nativního boxu ve všech prohlížečích stejně. Dalším rozdílem je, že umožňuje vložení HTML kódu, takže nejsme omezení jen na text. Dále umožňuje nastavit různé animační efekty. V aplikaci ji používám k validaci formulářů na straně klienta. [22]

```
swal({
  title: 'Chyba!',
  type: 'error',
  html: 'Heslo musí mít minimálně 7 znaků'
});
```

Výpis 1: Ukázka použití komponenty SweetAlert2

3.6.7 Nette.ajax.js 2.3

Tato knihovna není součástí Nette Frameworku. Knihovna implementuje AJAX požadavky na straně klienta. Někdy je třeba v aplikaci překreslit jen část webové stránky, tyto výřezy se nazývají snippety. V Latte se značí pomocí makra `snippet`. Pokud potřebuji, aby byl odkaz proveden pomocí AJAX, přidám odkazujícímu HTML elementu třídu „**ajax**“. Server pak rozliší, zda se jedná o AJAX požadavek. Pokud ano, překreslí jen danou část webové stránky. [23]



Obrázek 11: Ukázka použití komponenty SweetAlert2 pro validaci formuláře

3.6.8 NetteForms (součástí instalace Nette)

Slouží pro validaci pravidel na straně klienta, která jsou přenesena do HTML atributů „**data-nette-rules**“ vstupních polí formuláře. Validace se provádí při odchycení události **submit**, nikoliv už během vyplňování hodnot. Při objevení porušení pravidla se zobrazí nativní vyskakovací box prohlížeče s chybovou hláškou. Kód komponenty jsem upravil tak, že jsem tento box v aplikaci nahradil boxem z komponenty SweetAlert2.

3.6.9 Chart.js 2.7.3

Chart.js je open-source komponenta pro vykreslování responzivních grafů. Grafy se vykreslují do HTML tagů **canvas**. V aplikaci ji využívám pro vykreslení grafu s výdělky na nástěnce. [24]

3.6.10 Fio

Tato PHP komponenta umožňuje číst a zadávat platby u účtů vedených u Fio banky pomocí Fio API bankovníctví. Pro její použití ve frameworku Nette je třeba nainstalovat její rozšířenou verzi Fio for Nette. [25]

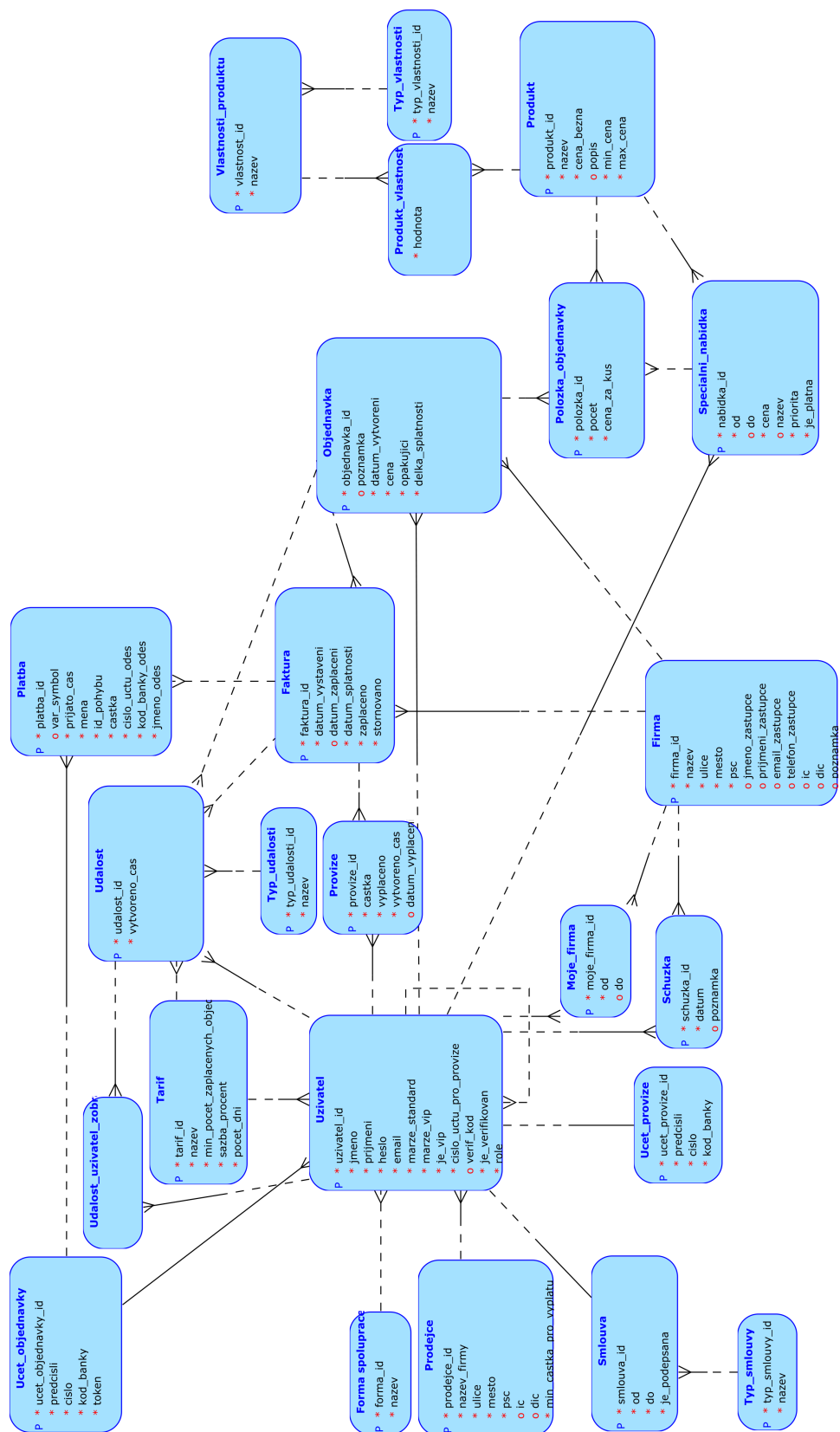
Shrnutí

V této kapitole jsme si přiblížili obchodní model pro aplikaci a několik důležitých technologií a komponent, které budou použité při samotné implementaci aplikace. Vždy jsem uvedl, proč jsem zvolil právě tyto technologie.

4 Analýza

V této kapitole si ukážeme strukturu tabulek uložených v databázi. Na dvou diagramech je také znázorněno, jakými vztahy jsou jednotlivé tabulky mezi sebou propojeny. Ve druhé části kapitoly si popíšeme první koncept vzhledu uživatelského rozhraní, podle kterého jsem se řídil při implementaci. Nakonec si projdeme samotnou implementaci aplikace, kde se dozvíte, jakým způsobem jsem jednotlivé technologie použil.

4.1 Doménový model



Obrázek 12: Konceptuální diagram

4.1.1 Lineární zápis

Vysvětlivky: **Primární klíč**, cizí klíč, *unikátní klíč*

Faktura (**faktura_id**, datum_vystaveni, datum_zaplaceni, datum_splatnosti, objednavka, firma, zaplaceno, stornovano)

Firma (**firma_id**, nazev, ulice, mesto, psc, jmeno_zastupce, prijmeni_zastupce, email_zastupce, telefon_zastupce, ic, dic, poznamka)

Forma_spoluprace (**forma_id**, nazev)

Moje_firma (**moje_firma_id**, od, do, uzivatel, firma)

Nabídka (**nabídka_id**, od, do, cena, nazev, produkt, priorita, uzivatel, je_platna)

Objednavka (**objednavka_id**, poznamka, datum_vytvoreni, cena, opakujici, firma, uzivatel, delka_splatnosti)

Platba (**platba_id**, var_symbol, ucet_objednavky, faktura, prijato_cas, mena, *id_pohybu*, castka, cislo_uctu_odes, kod_banky_odes, jmeno_odes)

Polozka_objednavky (**polozka_id**, pocet, cena_za_kus, produkt, objednavka, specialni_nabídka)

Prodejce (**prodejce_id**, nazev_firmy, ulice, mesto, psc, ic, dic, min_castka_pro_vyplatu)

Produkt (**produkt_id**, nazev, cena_bezna, popis, min_cena, max_cena)

Produkt_vlastnost (produkt_id, vlastnost_id, hodnota_vlastnosti)

Provize (**provize_id**, castka, vyplaceno, uzivatel, objednavka, stornovano, vytvoreno_cas, datum_vyplaceni)

Schuzka (**schuzka_id**, datum, poznamka, firma, uzivatel)

Smlouva (**smlouva_id**, od, do, uzivatel, je_podepsana, typ_smlouvy)

Tarif (**tarif_id**, nazev, min_pocet_zaplacenych_objednavek, sazba_procent)

Typ_smlouvy (**typ_smlouvy_id**, nazev)

Typ_udalosti (**typ_udalosti_id**, nazev)

Typ_vlastnosti (**typ_vlastnosti_id**, nazev)

Ucet_objednavky (**ucet_objednavky_id**, predcisli, cislo, kod_banky, token)

Ucet_provize (**ucet_provize_id**, predcisli, cislo, kod_banky, uzivatel)

Udalost (**udalost_id**, typ_udalosti, uzivatel_vytvoril, objednavka, faktura, tarif, vytvoreno_cas)

Udalost_uzivatel_zobraz (**udalost_uzivatel_zobraz_id**, udalost, uzivatel)

Uzivatel (**uzivatel_id**, jmeno, prijmeni, heslo, *email*, role, nadrazeny_uzivatel, marze_standard, marze_vip, je_vip, tarif, ucet_objednavky, forma_spoluprace, prodejce, verif_kod, je_verifikovan)

Vlastnost_produkту (**vlastnost_id**, nazev, jednotky, typ)

4.1.2 Datový slovník

Tabulky 3 až 26 popisují strukturu jednotlivých tabulek uložených v databázi.

Tabulka 3: Datový slovník tabulky Faktura

Název	Typ	Velikost	Klíč	Null	Index	Popis
faktura_id	Int	11	PK	Ne	Ano	Identifikace
datum_vystaveni	Date	-	-	Ne	Ne	Datum vystavení
datum_zaplaceni	Date	-	-	Ano	Ne	Datum zaplacení
datum_splatnosti	Date	-	-	Ne	Ne	Datum splatnosti
objednavka	Int	11	FK	Ne	Ne	Objednávka
firma	Int	11	FK	Ne	Ne	Firma
zaplaceno	Tinyint	1	-	Ne	Ne	Zaplaceno
stornovano	Tinyint	1	-	Ne	Ne	Stornováno

Tabulka 4: Datový slovník tabulky Firma

Název	Typ	Velikost	Klíč	Null	Index	Popis
firma_id	Int	11	PK	Ne	Ano	Identifikace
nazev	Varchar	60	-	Ne	Ne	Název
ulice	Varchar	60	-	Ne	Ne	Ulice sídla
mesto	Varchar	60	-	Ne	Ne	Město sídla
psc	Int	11	-	Ne	Ne	PSČ sídla
jmeno_zastupce	Varchar	60	-	Ano	Ne	Jméno zástupce
prijmeni_zastupce	Varchar	60	-	Ano	Ne	Příjmení zástupce
email_zastupce	Varchar	255	-	Ano	Ne	Email zástupce
telefon_zastupce	Int	11	-	Ano	Ne	Telefon zástupce
ic	Varchar	30	-	Ano	Ne	IČ
dic	Varchar	30	-	Ano	Ne	DIČ
poznamka	Varchar	255	-	Ano	Ne	Poznámka

Tabulka 5: Datový slovník tabulky Forma_spoluprace

Název	Typ	Velikost	Klíč	Null	Index	Popis
forma_id	Int	11	PK	Ne	Ano	Identifikace
nazev	Varchar	60	-	Ne	Ne	Název

Tabulka 6: Datový slovník tabulky Moje_firma

Název	Typ	Velikost	Klíč	Null	Index	Popis
moje_firma_id	Int	11	PK	Ne	Ano	Identifikace
od	Date	-	-	Ne	Ne	Začátek platnosti
do	Date	-	-	Ano	Ne	Konec platnosti
uzivatel	Int	11	FK	Ne	Ne	Uživatel
firma	Int	11	FK	Ne	Ne	Firma

Tabulka 7: Datový slovník tabulky Nabídka

Název	Typ	Velikost	Klíč	Null	Index	Popis
nabidka_id	Int	11	PK	Ne	Ano	Identifikace
od	Datetime	-	-	Ne	Ne	Začátek platnosti
do	Datetime	-	-	Ano	Ne	Konec platnosti
produkt	Int	11	FK	Ne	Ne	Produkt
priorita	Int	11	-	Ne	Ne	Priorita
uzivatel	Int	11	FK	Ne	Ne	Uživatel
je_platna	Tinyint	1	-	Ne	Ne	Je platná

Tabulka 8: Datový slovník tabulky Objednavka

Název	Typ	Velikost	Klíč	Null	Index	Popis
objednavka_id	Int	11	PK	Ne	Ano	Identifikace
poznámka	Varchar	255	-	Ano	Ne	Poznámka
datum_vytvoreni	Date	-	-	Ne	Ne	Datum vytvoření
cena	Float	-	-	Ne	Ne	Cena celkem
opakujici	Tinyint	1	-	Ne	Ne	Je opakující se
firma	Int	11	FK	Ne	Ne	Firma
uzivatel	Int	11	FK	Ne	Ne	Uživatel
delka_splatnosti	Int	11	-	Ne	Ne	Délka splatnosti ve dnech

Tabulka 9: Datový slovník tabulky Platba

Název	Typ	Velikost	Klíč	Null	Index	Popis
platba_id	Int	11	PK	Ne	Ano	Identifikace
var_symbol	Varchar	10	-	Ano	Ne	Variabilní symbol
ucet_objednavky	Int	11	FK	Ano	Ne	Účet objednávky
faktura	Int	11	FK	Ne	Ne	Faktura
prijato_cas	Datetime	-	-	Ne	Ne	Čas přijetí
mena	Varchar	3	-	Ne	Ne	Měna
id_pohybu	Varchar	30	UK	Ne	Ne	ID pohybu
castka	Float	-	-	Ne	Ne	Částka
cislo_uctu_odes	Varchar	10	-	Ne	Ne	Číslo účtu odesílatele
kod_banky_odes	Varchar	4	-	Ne	Ne	Kód banky odesílatele
jmeno_odes	Varchar	60	-	Ne	Ne	Jméno odesílatele

Tabulka 10: Datový slovník tabulky Polozka_objednavky

Název	Typ	Velikost	Klíč	Null	Index	Popis
polozka_id	Int	11	PK	Ne	Ano	Identifikace
pocet	Int	11	-	Ne	Ne	Počet kusů
cena_za_kus	Float	-	-	Ne	Ne	Cena za kus
produkt	Int	11	FK	Ne	Ne	Produkt
objednavka	Int	11	FK	Ne	Ne	Objednávka
specialni_nabidka	Int	11	FK	Ano	Ne	Speciální nabídka

Tabulka 11: Datový slovník tabulky Prodejce

Název	Typ	Velikost	Klíč	Null	Index	Popis
prodejce_id	Int	11	PK	Ne	Ano	Identifikace
nazev_firmy	Varchar	60	-	Ne	Ne	Název firmy
ulice	Varchar	60	-	Ne	Ne	Ulice
mesto	Varchar	60	-	Ne	Ne	Město
psc	Int	11	-	Ne	Ne	PSČ
ic	Varchar	30	-	Ano	Ne	IČ
dic	Varchar	30	-	Ano	Ne	DIČ
min_castka_pro_vyplatu	Float	-	-	Ne	Ne	Mín. výplata

Tabulka 12: Datový slovník tabulky Produkt

Název	Typ	Velikost	Klíč	Null	Index	Popis
produkt_id	Int	11	PK	Ne	Ano	Identifikace
nazev	Varchar	60	-	Ne	Ne	Název
cena_bezna	Float	-	-	Ne	Ne	Běžná cena
popis	Varchar	255	-	Ne	Ne	Popis
min_cena	Float	-	-	Ne	Ne	Minimální cena
max_cena	Float	-	-	Ne	Ne	Maximální cena

Tabulka 13: Datový slovník tabulky Produkt_vlastnost

Název	Typ	Velikost	Klíč	Null	Index	Popis
produkt_id	Int	11	FK	Ne	Ne	Identifikace produktu
vlastnost_id	Int	11	FK	Ne	Ne	Identifikace vlastnosti
hodnota_vlastosti	Varchar	60	-	Ne	Ne	Hodnota vlastnosti prod.

Tabulka 14: Datový slovník tabulky Provize

Název	Typ	Velikost	Klíč	Null	Index	Popis
provize_id	Int	11	PK	Ne	Ano	Identifikace
castka	Float	-	-	Ne	Ne	Částka
vyplaceno	Tinyint	1	-	Ne	Ne	Vyplaceno
uzivatel	Int	11	FK	Ne	Ne	Uživatel
objednavka	Int	11	FK	Ne	Ne	Objednávka
stornovano	Tinyint	1	-	Ne	Ne	Stornováno
vytvoreno_cas	Datetime	-	-	Ne	Ne	Čas vytvoření
datum_vyplaceni	Date	-	-	Ano	Ne	Datum vyplacení

Tabulka 15: Datový slovník tabulky Schuzka

Název	Typ	Velikost	Klíč	Null	Index	Popis
schuzka_id	Int	11	PK	Ne	Ano	Identifikace
datum	Datetime	-	-	Ne	Ne	Datum
poznámka	Varchar	255	-	Ano	Ne	Poznámka
firma	Int	11	FK	Ne	Ne	Firma
uzivatel	Int	11	FK	Ne	Ne	Uživatel

Tabulka 16: Datový slovník tabulky Smlouva

Název	Typ	Velikost	Klíč	Null	Index	Popis
smlouva_id	Int	11	PK	Ne	Ano	Identifikace
od	Datetime	-	-	Ne	Ne	Začátek platnosti
do	Datetime	-	-	Ano	Ne	Konec platnosti
uzivatel	Int	11	FK	Ne	Ne	Uživatel
je_podepsana	Tinyint	1	-	Ne	Ne	Je podepsaná
typ_smlouvy	Int	11	FK	Ne	Ne	Typ smlouvy

Tabulka 17: Datový slovník tabulky Tarif

Název	Typ	Vel.	Klíč	Null	Index	Popis
tarif_id	Int	11	PK	Ne	Ano	Identifikace
nazev	Varchar	60	-	Ne	Ne	Název
min_pocet_zaplacených_objednavek	Int	11	-	Ne	Ne	Min. počet zaplacených obj.
sazba_procent	Int	11	-	Ne	Ne	Sazba v procentech

Tabulka 18: Datový slovník tabulky Typ_smlouvy

Název	Typ	Velikost	Klíč	Null	Index	Popis
typ_smlouvy_id	Int	11	PK	Ne	Ano	Identifikace
nazev	Varchar	60	-	Ne	Ne	Název

Tabulka 19: Datový slovník tabulky Typ_udalosti

Název	Typ	Velikost	Klíč	Null	Index	Popis
typ_udalosti_id	Int	11	PK	Ne	Ano	Identifikace
nazev	Varchar	60	-	Ne	Ne	Název

Tabulka 20: Datový slovník tabulky Typ_vlastnosti

Název	Typ	Velikost	Klíč	Null	Index	Popis
typ_vlastnosti_id	Int	11	PK	Ne	Ano	Identifikace
nazev	Varchar	60	-	Ne	Ne	Název

Tabulka 21: Datový slovník tabulky Ucet_objednavky

Název	Typ	Velikost	Klíč	Null	Index	Popis
ucet_objednavky_id	Int	11	PK	Ne	Ano	Identifikace
predcisli	Varchar	6	-	Ne	Ne	Předčísli
cislo	Varchar	10	-	Ne	Ne	Číslo
kod_banky	Varchar	4	-	Ne	Ne	Kód banky
token	Varchar	64	-	Ne	Ne	Token pro přístup k API

Tabulka 22: Datový slovník tabulky Ucet_provize

Název	Typ	Velikost	Klíč	Null	Index	Popis
ucet_provize_id	Int	11	PK	Ne	Ano	Identifikace
predcisli	Varchar	6	-	Ne	Ne	Předčísli
cislo	Varchar	10	-	Ne	Ne	Číslo
kod_banky	Varchar	4	-	Ne	Ne	Kód banky
uzivatel	Int	11	FK	Ne	Ne	Uživatel

Tabulka 23: Datový slovník tabulky Udalost

Název	Typ	Velikost	Klíč	Null	Index	Popis
udalost_id	Int	11	PK	Ne	Ano	Identifikace
typ_udalosti	Int	11	FK	Ne	Ne	Typ události
uzivatel_vytvoril	Int	11	FK	Ne	Ne	Uživatel, který ji vytvořil
objednavka	Int	11	FK	Ano	Ne	Objednávka
faktura	Int	11	FK	Ano	Ne	Faktura
tarif	Int	11	FK	Ano	Ne	Tarif
vytvoreno_cas	Datetime	-	-	Ne	Ne	Čas vytvoření

Tabulka 24: Datový slovník tabulky Udalost_uzivatel_zobraz

Název	Typ	Velikost	Klíč	Null	Index	Popis
udalost_uzivatel_zobraz_id	Int	11	PK	Ne	Ano	Identifikace
udalost	Int	11	FK	Ne	Ne	Událost
uzivatel	Int	11	FK	Ne	Ne	Uživatel

Tabulka 25: Datový slovník tabulky Uzivatel

Název	Typ	Velikost	Klíč	Null	Index	Popis
uzivatel_id	Int	11	PK	Ne	Ano	Identifikace
jmeno	Varchar	60	-	Ne	Ne	Jméno
prijmeni	Varchar	60	-	Ne	Ne	Příjmení
heslo	Varchar	255	-	Ne	Ne	Heslo
email	Varchar	255	UK	Ne	Ne	Email
role	Varchar	60	-	Ne	Ne	Role
nadrazeny_uzivatel	Int	11	FK	Ano	Ne	Nadřazený uživatel
marze_standard	Float	-	-	Ne	Ne	Marže, pokud už. není VIP
marze_vip	Float	-	-	Ne	Ne	Marže, pokud už. je VIP
je_vip	Tinyint	1	-	Ne	Ne	Je VIP
tarif	Int	11	FK	Ne	Ne	Tarif
ucet_objednavky	Int	11	FK	Ne	Ne	Účet objednávky
forma_spoluprace	Int	11	FK	Ano	Ne	Forma spolupráce
prodejce	Int	11	FK	Ne	Ne	Prodejce
verif_kod	Varchar	20	-	Ano	Ne	Kód pro verifikaci
je_verifikovan	Tinyint	1	-	Ne	Ne	Je verifikován

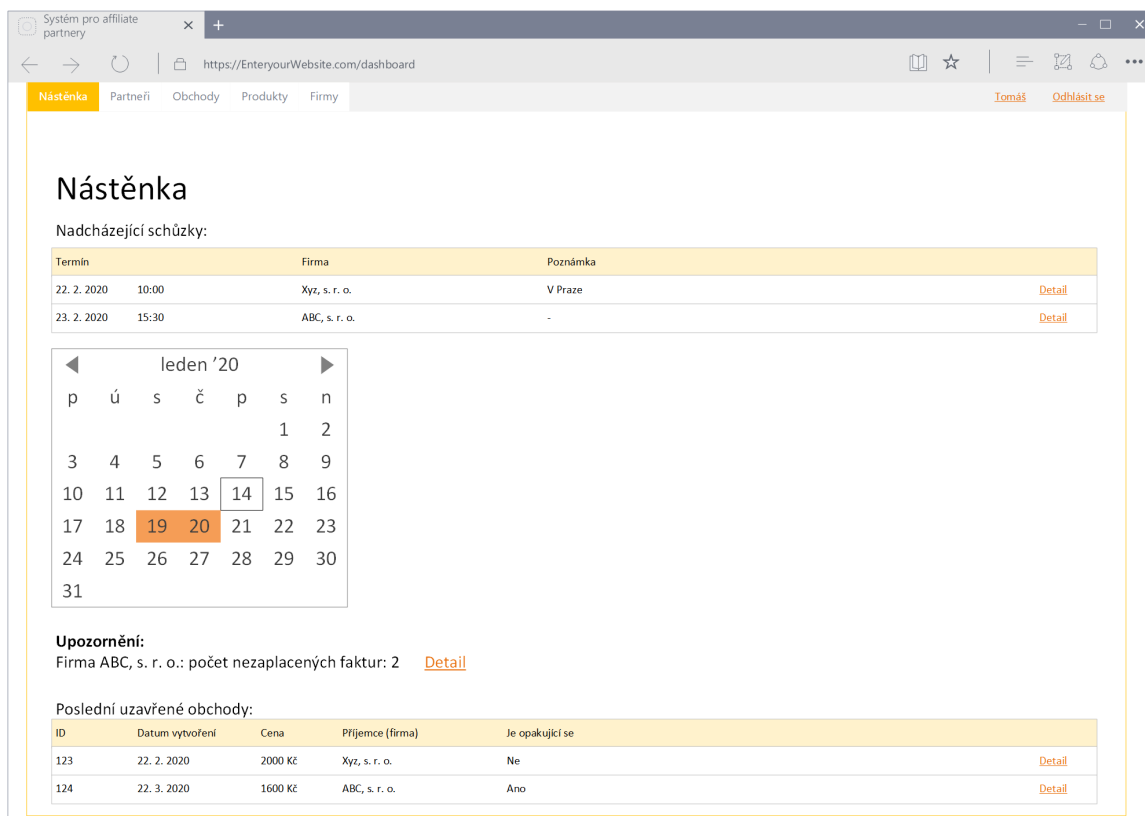
Tabulka 26: Datový slovník tabulky Vlastnost_produkту

Název	Typ	Velikost	Klíč	Null	Index	Popis
vlastnost_id	Int	11	PK	Ne	Ano	Identifikace
nazev	Varchar	60	-	Ne	Ne	Název
jednotky	Varchar	60	-	Ne	Ne	Jednotky
typ	Int	11	FK	Ne	Ne	Typ

4.2 Koncept vzhledu uživatelského rozhraní

Při implementaci aplikace jsem vycházel z návrhu drátěného modelu uživatelského rozhraní, pro jehož tvorbu jsem použil program Microsoft Visio.

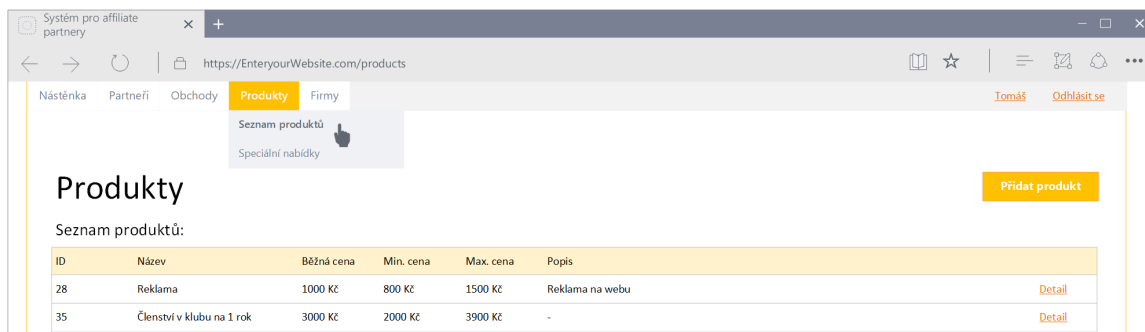
Výchozím bodem aplikace je nástěnka (14), na kterou se uživatel dostane hned po přihlášení. Měla by sloužit jako hlavní přehled. Obsahuje kalendář schůzek uživatele s firmami. Další důležitou součástí nástěnky je upozornění na nezaplacené faktury.



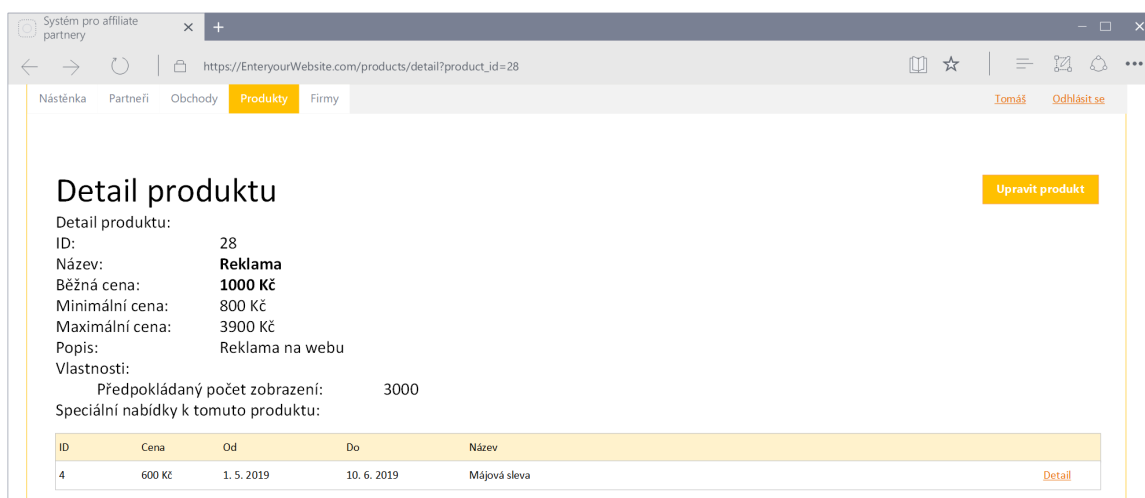
Obrázek 14: Koncept vzhledu nástěnky

Z hlavní nabídky se uživatel dostane na přehledy (tabulky) partnerů, objednávek, produktů, firem a schůzek. Přehled produktů je znázorněn na obrázku 15. V přehledu je většinou tlačítko pro přidání nové entity.

Z přehledu se dále dostane do detailu jednotlivých entit (znázorněno na obrázku 16), odkud se dá přejít na formulář pro úpravu dané entity (znázorněno na obrázku 18). U některých entit se na stránce detailu nachází také seznam souvisejících entit, například speciální nabídky u produktu.



Obrázek 15: Koncept vzhledu seznamu produktů



Obrázek 16: Koncept vzhledu detailu produktu

Systém pro affiliate partnery

https://EnteryourWebsite.com/products/edit/?productid=1

Nástěnka Partneři Obchody **Produkty** Firmy

Tomáš Odhlásit se

Editace produktu

Název

Reklama

Popis

Reklama na webu

Běžná cena

1000 Kč

Minimální cena

800 Kč

Maximální cena

1500 Kč

Přidat další vlastnosti

Uložit produkt

Obrázek 17: Koncept vzhledu editace produktu

4.3 Implementace aplikace

4.3.1 Aplikační vrstva

Většina aplikační logiky je řešena v presenterech, kde pro každou sekci aplikace existuje jeden presenter. Tyto presentery jsou zde odvozené třídy z rodičovské třídy **BasePresenter**.

Při spuštění aplikační sekce se zavolá v daném presenteru nejprve metoda **startup**. Uvnitř metody vždy kontroluji, zda má uživatel dostatečná oprávnění pro to, aby mohl do dané sekce vstoupit. Pokud jej nemá, můžu jej přesměrovat do jiné sekce (např. stránku s přihlašovacím formulářem), nebo nechat zavolat výjimku.

Před vykreslením šablony se volá metoda **beforeRender()**. V mé aplikaci má význam zejména při vykreslení položek v uživatelském menu. Například administrátor vidí jiné položky než ostatní přihlášení prodejci. Nepřihlášenému uživateli není třeba vykreslovat žádné položky. Další využití této metody je předání titulku stránky do šablony, který je stejný pro všechny podsektce.

V odvozených presenterech se obvykle nachází metoda **renderDefault()**. Ta slouží například pro naplnění dat do šablony, a to v případě, že se uživatel nachází v kořenu sekce. Pokud se nachází v nějaké podsektci, pro naplnění dat do ní je třeba použít metodu, jejíž název je tvořen prefixem „**render**“, za nímž následuje název podsektce. V některých případech, kdy se nic nevykresluje, je postačující místo metody **render<Podsektce>()** použít alternativní metodu s prefixem „**action**“. Metoda **action<Podsektce>()** se volá vždy ještě před metodou **render<Podsektce>()** a umožňuje rozhodnout, zda se metoda **render<Podsektce>()** bude volat, či nikoliv.

4.3.2 Formuláře

Předtím, než budeme vykreslovat formulář do šablony, musíme jej vytvořit v presenteru pomocí tzv. továrny na komponentu formuláře. Továrna umožňuje vytvářet komponenty ve chvíli, kdy jej opravdu potřebujeme. Továrnu vytvoříme tak, že metodě dáme název **createComponent<>(Name)**, kde **Name** bude název komponenty, kterou chceme vytvořit. Komponentou se myslí potomek třídy **Control**, tedy se nemusí jednat jen o formulář.

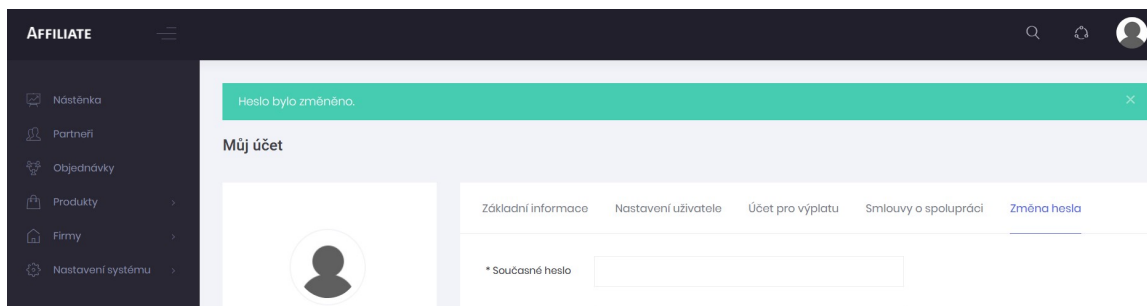
Uvnitř metody (továrny) vytvoříme nový objekt třídy **UI/Form**. Na tento objekt pak voláme metody pro přidávání formulářových prvků, například **addText(string|int \$name, \$label=null)**. Na tyto prvky pak můžeme volat metodu **setRequired()**, kterou určíme, zda musí být prvek vyplněn. Dále metodu **addRule()**, pomocí níž přidáme validační pravidlo, například maximální hodnotu, nebo metodu **setDefaultValue()**, pomocí níž nastavíme předvyplněnou hodnotu. Výchozí hodnoty můžeme u celého formuláře nastavit také najednou pomocí metody **setDefaults()**.

V Latte šabloně je třeba formulář vykreslit, to můžeme udělat pomocí makra **{control <Name>}** nebo jednotlivé prvky můžeme vykreslit manuálně pomocí HTML.

4.3.3 Flash zprávy

Někdy je třeba informovat uživatele o výsledku nějaké operace, například o úspěšné nebo neúspěšné změně hesla. K tomu v aplikaci používám tzv. flash zprávy, které zůstanou v šabloně k dispozici i pokud dojde k přesměrování. Tyto zprávy zůstanou v šabloně také, pokud uživatel během tří sekund stránku obnoví nebo přejde do jiné sekce této webové aplikace.

Pro vytvoření zprávy stačí v presenteru jen zavolat metodu **flashMessage()**, kde prvním parametrem je text zprávy. Druhým a nepovinným parametrem je typ zprávy, například „**success**“. Typem zprávy pak můžeme ovlivnit vzhled zprávy, třeba barvu pozadí nebo ikonu. V šabloně ke zprávám přistupujeme pomocí anonymních objektů z proměnné **\$flashes**.



Obrázek 18: Flash zpráva oznamující úspěšnou změnu hesla

```
<div n:foreach="$flashes as $flash" n:class="alert, alert-dismissible, fade,
    show, 'alert-' . $flash->type">
  {$flash->message}
  <button type="button" class="close" data-dismiss="alert"></button>
</div>
```

Výpis 2: Výpis flash zpráv do šablony

4.3.4 Zpracování AJAX požadavků

Naplňování dat do tabulek probíhá pomocí AJAX požadavku, kde server vrací JSON soubor s daty. Pro zpracování požadavku je třeba v presenteru vytvořit metodu **handle<Name>()**, kde **Name** je název signálu. V URL je tento název předán parametrem „**do**“. Pro odeslání odpovědi uvnitř metody zavoláme na presenter metodu **sendResponse()** s parametrem třídy **JsonResponse**.

Pokud potřebujeme zjistit, zda byla URL načtena pomocí AJAX, zavoláme na presenter metodu **isAjax()**. To se může hodit, když chceme po provedení požadavku překreslit jen část webové stránky. Pro její překreslení musí být tato část/komponenta v šabloně označena makrem **{snippet <Name>}**, kde **Name** je název komponenty. Pak už jen na presenter zavoláme

metodu **redrawControl()**, kde do parametru uvedeme název komponenty. Tímto způsobem řeším v aplikaci například změnu stavu (povoleno/zakázáno) u speciálních nabídek produktu. Při kliknutí na tlačítko (povoleno/zakázáno) se znovu načte komponenta s nabídkami (**,offers‘**), kde se změní text a barva tlačítka a komponenta produktu (**,productDetail‘**), uvnitř které se změní „Moje aktuální cena“ na právě přepočítanou cenu.

4.3.5 Odesílání emailů

Ve své aplikaci potřebuji odesílat emaily, pokud někdo zaregistruje nového uživatele. Součástí takového emailu je pak odkaz, který obsahuje v parametru verifikační kód. Pokud na odkaz uživatel klikne, dostane se na stránku, v níž vytvoří heslo pro svůj nový účet a dalším krokem se jeho účet aktivuje. Funkce odesílání emailů je tak pro tuto aplikaci klíčová.

Implementace pomocí třídy **Nette/Mail** je jednoduchá. Vytvoříme si nejprve novou instanci třídy **Nette/Mail/Message** a pak na ní voláme příslušné metody. První je metoda **setFrom()**, která nastaví odesílatele zprávy. Příjemce přidáváme pomocí metody **addTo()**, kde parametrem je emailová adresa. Předmět nastavíme pomocí metody **setSubject()**. HTML obsah zprávy nastavíme metodou **setHtmlBody()**, které předáme parametrem HTML kód. Pro odeslání je třeba vytvořit instanci třídy **Nette/Mail/SendmailMailer** a zavolat na ní metodu **send()** s parametrem instance **Nette/Mail/Message**, kterou jsme si již vytvořili. **SendmailMailer** používá PHP funkci **mail()**.

4.3.6 Generování PDF souborů

V aplikaci potřebuji vytvářet PDF soubory automaticky podle předaných parametrů například pro faktury k objednávkám. Protože Nette framework žádnou možnost nenabízí, je pro tento účel vhodné využít knihovnu třetí strany.

Generování PDF souborů provádím pomocí knihovny **Mpdf**. Ta jej generuje z HTML kódu a dokáže jej také nastylovat podle externího CSS kódu.

Prvním krokem pro vytvoření PDF souboru je vytvoření instance třídy **Mpdf**. Dále mám vytvořenou Latte šablonu, kterou jen naplním daty (pro fakturu to jsou položky objednávky, adresy, ceny apod.) a pak ji předám jako parametr metodě **WriteHTML()**, kterou volám na instanci **Mpdf**. Můžu také nastavit například patičku, která bude u každé stránky, například s číslem strany, a to voláním metody **setFooter()**. Výstupní soubor získám jednoduše zavoláním metody **Output()**, kde jako parametr uvedu název souboru.

4.3.7 Vrstva přístupu k databázi

Tato aplikace je již velmi rozsáhlá a je proto důležité se při jejím vývoji snažit, aby byl kód co nejpřehlednější. Jeden ze způsobů, jak přehlednost zlepšit, je vytvořit a oddělit od aplikační vrstvy metody, které přistupují k databázi, a to vytvořením oddělené vrstvy. Pokud v budoucnu upravíme například název sloupce v databázi, nebudeme muset upravovat dotaz pro získání dat

z tabulky, který se nachází několikrát v různých presenterech. Místo toho bude stačit jen upravit jednu metodu v modelu. Bude tak navíc menší šance, že se někde spleteme a vyrobíme chybu, které si třeba ani nevšimneme.

V adresáři `app/model` se nachází soubory s třídami, které pracují s databází. Tyto soubory a třídy uvnitř mají příponu „Manager“. Instance těchto tříd pak vytvářím v konstruktorech presenterů a z nich pak na tyto instance volám příslušné metody pro práci s databází. V jejich konstruktorech se vždy do privátní proměnné `$database` přiřadí databázový kontext (instance třídy **Nette/Database/Context**), pomocí něhož pak vytváříme dotazy na databázi.

Nette Database nám velmi zjednodušuje práci s databází, protože nemusíme psát SQL dotazy. Jednoduše na proměnnou s databázovým kontextem (`$database`) voláme postupně metody, které zastupují jednotlivé části SQL dotazu.

Tvorbu dotazu začneme metodou **table()**, kde parametrem předáme název tabulky. Metoda vrátí objekt **Selection**, který obsahuje všechny řádky tabulky. Jednotlivé řádky jsou instancemi třídy **ActiveRow** a data z nich můžeme přímo číst. Potřebujeme-li výběr omezit stejně jako bychom v SQL použili podmínku `WHERE`, zavoláme na objekt **Selection** metodu **where()**, se zněním podmínky uvnitř. Tato metoda dokonce může obsahovat i klíč z jiné tabulky, pokud jsou tyto tabulky mezi sebou propojeny. Nepotřebujeme tak metodu pro `JOIN` (Nette Database tuto metodu neobsahuje). Mezi další podobné metody se řadí také například **whereOr()** pro spojení více podmínek pomocí `OR`, dále metoda **order()** pro nastavení `ORDER BY`, dále **select()** pro nastavení vrácených sloupců, **page()** pro stránkování, **group()** pro nastavení `GROUP BY` a **having()** pro nastavení `HAVING`.

Chceme-li vložit záznamy, voláme metodu **insert()**, ta umožňuje vložení více záznamů najednou, předáme-li do ní pole záznamů. Pro úpravu voláme metodu **update()** a pro smazání metodu **delete()**. Obě metody vrací počet ovlivněných řádků.

Názvy metod jsou jednoduše zapamatovatelné, neboť jsou často stejné jako jejich zápis v SQL dotazu.

Při vytváření dotazu je třeba vždy myslet na to, že stejně jako u psaní klasického SQL dotazu, i u Nette Database může dojít k SQL injection. Abychom tomu zabránili, musíme místo skládání dotazů dosazovat hodnoty přes parametry.

4.3.8 Šablonovací systém Latte

Součástí frameworku Nette je Latte [26]. Je to systém, díky kterému lze HTML kód bezpečně kombinovat s jazykem PHP. Šablony jsou překládány na optimalizovaný kód v jazyce PHP.

Je to první PHP engine, které využívá kontextově sensitivní escapování (tzn. automatický převod znaků, které mají v kontextu speciální význam na jiné odpovídající sekvence podle toho, ve které části dokumentu se makro nachází) k předcházení útoků typu Cross Site Scripting a kontrolu odkazů (zda adresa v attributech `src` a `href` odkazuje na URL s protokolem `HTTP`).

Existuje několik maker, která nahradí kód, který bychom jinak psali pomocí PHP, zapisujeme je dovnitř složených závorek. Do latte můžeme doprogramovat také vlastní makra.

Pokud chceme nějaký text modifikovat (například, aby byl psán velkými písmeny), můžeme k tomu použít tzv. filtry. Opět jich existuje několik standartních, stejně tak si můžeme vytvořit své vlastní. Filtry se zapisují za svislítko (kde před svislítkem je text, na který má být filtr aplikován).

O chybách a překlepech v kódu jsme informováni v prohlížeči, protože Latte spolupracuje s ladícím nástrojem Tracy.

Shrnutí

Nyní už víte dopodrobna, jak vypadá databáze pro tuto aplikaci. Prohlédli jste si také, z jakého návrhu vzhledu jsem při implementaci vycházel. Nakonec, v nejdůležitější části kapitoly, jste se dozvěděli, jak jsem při implementaci aplikace použil nejdůležitější technologie, tedy zejména framework Nette, formuláře a AJAX volání. Dále víte, že aplikace přistupuje k databázi pomocí databázového kontextu z frameworku Nette. Také jste se dozvěděli, že pro výpis šablon je používán systém Latte.

5 Testování aplikace

Před nasazením každé aplikace do ostrého provozu je vhodné ji co nejvíce otestovat a odhalit tak co nejvíce problémů a chyb včas. Kromě toho je také dobré vědět, jak budou aplikaci rozumět její budoucí uživatelé, a zda bude nutné vytvořit k aplikaci nápovědu nebo návod.

5.1 Způsob testování

Testování aplikace se zúčastnili čtyři lidé. Na začátku jsem každému vysvětlil princip a účel aplikace, poté jsme prošli všechny položky hlavní nabídky, jako je nástěnka, produkty apod., a to jak v prostředí administrátora, tak i prodejce. Požádal jsem je, aby si vyzkoušeli různé funkce. Mezi nimi byly například registrace nového účtu partnera, vytvoření produktu, objednávky a faktury, nebo naplánování schůzky. V průběhu jsem zjišťoval, do jaké míry aplikaci účastníci testování rozumí.

5.2 Průběh testování

V průběhu testování se ukázalo, že pro účastníky bylo nejtěžší pochopit samotný princip aplikace. Někteří totiž očekávali, že aplikace má sloužit také kupujícím. Všichni účastníci ale nakonec pochopili, že tomu tak není. Jako dostatečně přehledné a jednoduché bylo hodnoceno grafické rozhraní aplikace. Jednomu účastníkovi vadilo, že při přidávání faktur k objednávce nevidí potvrzovací zprávu. Ta se sice objeví, ale v horní části stránky. Na malé obrazovce je třeba se vrátit nahoru, aby bylo možné si zprávu prohlédnout. Tento problém však považuji za drobnost. Bohužel ale někteří účastníci narazili na větší chybu, která znemožňovala vyzkoušet některé funkce. Kvůli ní nebylo možné dostat se do nastavení některých uživatelských účtů. Další závažnou chybou bylo, že objednávku bylo možné vytvořit i když firma byla přiřazená uživateli pro jiné časové rozpětí. Obě závažné chyby jsem již vyřešil.

5.3 Doporučení dalšího rozšíření

Po dokončení testování jsem dostal od účastníků také návrhy na přidání nových funkcí. Jsou mezi nimi tyto:

1. Možnost přidání obrázku k produktu
2. Žebříček nejlepších partnerů
3. Odesílání faktur firmě na email přímo z aplikace
4. Informační zprávy přizpůsobit tak, aby byly vždy na viditelném místě

K těmto návrhům přidávám ještě své vlastní návrhy:

1. Vylepšit detail produktu o zobrazení počtu prodejů, případně seznamu faktur (objednávek), na kterých se položka produktu nachází
2. Skladové zásoby
3. Generování detailních statistik do CSV souboru
4. Pokročilé vyhledávání napříč všemi daty
5. Další přehledy na nástěnce – nejprodávanější produkty, všechny speciální nabídky
6. Možnost deaktivace produktu (aby nemohl být prodáván)

5.4 Průvodce aplikací

Přestože se zdá, že se všichni účastníci v aplikaci zorientovali, rozhodl jsem se pro tvorbu jednoduché nápovědy k aplikaci. Pro rychlejší a snadnější pochopení fungování aplikace budoucím uživatelům jsem vytvořil jednoduchého průvodce jednotlivými stránkami, kterého lze spustit po kliknutí na ikonu otazníku v pravém horním rohu. Tuto funkci jsem naimplementoval pomocí komponenty Intro.js.

Shrnutí

Testování aplikace hodnotím jako velmi přínosné. Pomohlo odhalit několik závažných chyb, které by v případě nasazení aplikace do produkčního režimu ohrožovaly její chod. Tyto chyby jsem následně opravil. Od účastníků testování jsem dostal několik návrhů na přidání nových funkcí. Aby se budoucí uživatelé v aplikaci rychleji zorientovali, vytvořil jsem jednoduchého průvodce aplikací, kterého mohou kdykoliv spustit kliknutím na ikonu otazníku v pravém horním rohu aplikace.

6 Závěr

Cíle práce byly splněny a dosažené výsledky otestovány. V závěru si dovolím porovnat svou aplikaci s podobným, existujícím projektem. Pokud bychom brali nástroj ProAffil (viz kapitola 2) jako referenční projekt (berme jej tak i přesto, že účel tohoto nástroje se od účelu mé aplikace poněkud liší, protože jsem žádný podobnější projekt nepoznal), moje aplikace se zdá být rozsáhlejší. Například nástěnka je u mé aplikace propracovanější, dokáže upozornit uživatele na faktury, kterým se blíží konec splatnosti, ukazuje přehled událostí v posledních dnech, a pokud si uživatel naplánuje schůzku s firmou, s níž chce uzavřít obchod, uvidí ji v kalendáři. Na druhou stranu má aplikace postrádá detailnější statistiky.

V rámci mé bakalářské práce jsem shrnul, jaké existují provizní nástroje pro podporu prodeje. Dále se mi podařilo navrhnout, naimplementovat a otestovat aplikaci, jejím účelem je podpořit prodeje produktů firmy pomocí obchodních zástupců. V práci jsem popsal technologie, které byly při tvorbě aplikace použity. Důležitou částí práce je také popis testování aplikace, které pomohlo odhalit chyby a přineslo návrhy na přidání nových funkcí. Aplikace je podle mého názoru dotvořená natolik, aby ji bylo možno skutečně používat. Všechny podstatné problémy, na které jsme při testování narazili, jsem již opravil.

Literatura

- [1] *Manutan partnerský program*. [online]. 2009 [cit. 20.03.2019]. Dostupné z: <https://www.manutan-partner.cz/>
- [2] *Affiliate*. [online]. [cit. 24.02.2018]. Dostupné z: <https://demo.proaffil.cz/>
- [3] *proaffil.cz*. [online]. 2017 [cit. 24.02.2018]. Dostupné z: <http://proaffil.cz/>
- [4] *Ceník / AffilBox*. [online]. 2011 [cit. 12.03.2019]. Dostupné z: <https://www.affilbox.cz/cenik/>
- [5] *AffilBox - Shoptet doplňky*. [online]. [cit. 12.03.2019]. Dostupné z: <https://doplňky.shoptet.cz/affilbox>
- [6] *Provizní systém - Shoptet doplňky*. [online]. [cit. 12.03.2019]. Dostupné z: <https://doplňky.shoptet.cz/provizni-system>
- [7] *In-house Affiliate Program Hosting / Post Affiliate Pro*. [online]. 2004 [cit. 12.03.2019]. Dostupné z: <https://www.postaffiliatepro.com/pricing/>
- [8] *Doporuč přáteli a získejte každý 500 Kč! / SmartEmailing*. [online]. 2018 [cit. 12.03.2019]. Dostupné z: <https://www.smartemailing.cz/doporuc-priteli-a-ziskejte-kazdy-500-kc/>
- [9] *Cashback portál Tipli - Získejte peníze z online nákupů / Tipli*. [online]. 2019 [cit. 12.03.2019]. Dostupné z: <https://www.tipli.cz/>
- [10] *ING Bank - Peníze na správném místě*. [online]. [cit. 12.03.2019]. Dostupné z: <https://www.ing.cz/files/cz/sporeni/podminky/pravidla-kampane-doporuc-ing-konto-cervenec.pdf>
- [11] *Bootstrap – Wikipedie*. [online]. [cit. 12.03.2019]. Dostupné z: <https://cs.wikipedia.org/wiki/Bootstrap>
- [12] *ASP.NET vs PHP - Top 8 Most Valuable Differences To Learn. Best Online Training & Video Courses / eduCBA*. [online]. [cit. 12.03.2019]. Dostupné z: <https://www.educba.com/asp-net-vs-php/>
- [13] *KeenThemes.com - Bootstrap Dashboard Themes & Templates*. [online]. Copyright © Keenthemes. [cit. 04.04.2019]. Dostupné z: <https://keenthemes.com/>
- [14] *Bootstrap – Wikipedie*. [online]. [cit. 12.03.2019]. Dostupné z: <https://cs.wikipedia.org/wiki/Bootstrap>
- [15] *Nette framework*. [online]. Dostupné z: <https://nette.org/>

- [16] *Differences between MVC and MVP for Beginners - CodeProject*. [online]. [cit. 04.04.2019]. Dostupné z: <https://www.codeproject.com/Articles/288928/Differences-between-MVC-and-MVP-for-Beginners>
- [17] *Metronic / Local Data. KeenThemes.com - Bootstrap Dashboard Themes & Templates*. [online]. [cit. 04.04.2019]. Dostupné z: <https://keenthemes.com/metronic/preview/?page=crud/metronic-datatable/base/data-local&demo=default>
- [18] *FullCalendar - JavaScript Event Calendar*. [online]. 2019 [cit. 04.04.2019]. Dostupné z: <https://fullcalendar.io/>
- [19] *Bootstrap TouchSpin. Virtuosoft*. [online]. 2013 [cit. 04.04.2019]. Dostupné z: <https://www.virtuosoft.eu/code/bootstrap-touchspin/>
- [20] *bootstrap-datepicker*. [online]. 2016 [cit. 04.04.2019]. Dostupné z: <https://bootstrap-datepicker.readthedocs.io/en/latest/index.html>
- [21] *bootstrap-select*. [online]. Dostupné z: <https://developer.snapappointments.com/bootstrap-select/examples/>
- [22] *SweetAlert2 - a beautiful, responsive, customizable and accessible (WAI-ARIA) replacement for JavaScript's popup boxes*. [online]. Dostupné z: <https://sweetalert2.github.io/>
- [23] *vojtech-dobes/nette.ajax.js / Componette*. [online]. 2015 [cit. 04.04.2019]. Dostupné z: <https://componette.com/vojtech-dobes/nette.ajax.js/>
- [24] *Chart.js / Open source HTML5 Charts for your website*. [online]. Dostupné z: <https://www.chartjs.org/>
- [25] *GitHub - h4kuna/fio: Read and send payment order for FIO bank*. [online]. 2019 [cit. 04.04.2019]. Dostupné z: <https://github.com/h4kuna/fio>
- [26] *Latte / Nette Framework*. [online]. [cit. 06.04.2019]. Dostupné z: <https://latte.nette.org/cs/guide>